mitp

François Chollet

J.J. Allaire

Deep Learning mit R und Keras

Das Praxis-Handbuch

von Entwicklern von Keras und RStudio

Inhaltsverzeichnis

Vorwort 13

Einleitung 15

Danksagungen 19

Über die Autoren 20

Teil I Grundlagen des Deep Learnings 21

1 Was ist Deep Learning? 23

1.1 Künstliche Intelligenz, Machine Learning und Deep Learning 23

1.1.1 Künstliche Intelligenz 24

1.1.2 Machine Learning 24

1.1.3 Die Repräsentation anhand der Daten erlernen 26

1.1.4 Das »Deep« in Deep Learning 29

1.1.5 Deep Learning in drei Diagrammen erklärt 31

1.1.6 Was Deep Learning heute schon leisten kann 33

1.1.7 Schenken Sie dem kurzfristigen Hype keinen Glauben 34

1.1.8 Das Versprechen der KI 35

1.2 Bevor es Deep Learning gab: eine kurze Geschichte des Machine Learnings 36

1.2.1 Probabilistische Modellierung 37

1.2.2 Die ersten neuronalen Netze 37

1.2.3 Kernel-Methoden 38

1.2.4 Entscheidungsbäume, Random Forests und Gradient Boosting Machines 40

1.2.5 Zurück zu neuronalen Netzen 41

1.2.6 Das Besondere am Deep Learning 42

1.2.7 Der Stand des modernen Machine Learnings 43

1.3 Warum Deep Learning? Und warum jetzt? 44

1.3.1 Hardware 45

1.3.2 Daten 46

1.3.3 Algorithmen 46

1.3.4 Eine neue Investitionswelle 47

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

1.3.5 Die Demokratisierung des Deep Learnings 48
1.3.6 Bleibt es so? 48

2 Bevor es losgeht: die mathematischen Bausteine eines neuronalen Netzes 51
2.1 Ein erster Blick auf ein neuronales Netz 51
2.2 Datenrepräsentationen 56
2.2.1 Skalare (0-D-Tensoren) 57
2.2.2 Vektoren (1-D-Tensoren) 57
2.2.3 Matrizen (2-D-Tensoren) 58
2.2.4 3-D-Tensoren und höherdimensionale Tensoren 58
2.2.5 Die wichtigsten Attribute 58
2.2.6 Bearbeiten von Tensoren in R 60
2.2.7 Datenstapel 60
2.2.8 Beispiele für Datentensoren aus der Praxis 61
2.2.9 Vektordaten 61
2.2.10 Zeitreihen oder sequenzielle Daten 62
2.2.11 Bilddaten 62
2.2.12 Videodaten 63
2.3 Das Getriebe von neuronalen Netzen: Tensoroperationen 64
2.3.1 Elementweise Operationen 65
2.3.2 Operationen mit Tensoren unterschiedlicher Dimensionalität 66
2.3.3 Tensorprodukt 66
2.3.4 Tensoren umformen 69
2.3.5 Geometrische Interpretation von Tensoroperationen 71
2.3.6 Eine geometrische Interpretation des Deep Learnings 72
2.4 Der Antrieb von neuronalen Netzen: gradientenbasierte Optimierung 73
2.4.1 Was ist eine Ableitung? 74
2.4.2 Ableitung einer Tensoroperation: der Gradient 76
2.4.3 Stochastisches Gradientenabstiegsverfahren 77
2.4.4 Ableitungen verketten: der Backpropagation-Algorithmus . . 80
2.5 Zurück zum ersten Beispiel 81
2.6 Zusammenfassung Kapitel 2 84

3 Einführung in neuronale Netze 85
3.1 Aufbau eines neuronalen Netzes 86
3.1.1 Layer: Bausteine des Deep Learnings 86
3.1.2 Modelle: vernetzte Layer 88

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

3.1.3 Verlustfunktionen und Optimierer: Konfiguration des Lernvorgangs 88
3.2 Einführung in Keras 89
3.2.1 Keras, TensorFlow, Theano und CNTK 91
3.2.2 Keras installieren 92
3.2.3 Mit Keras entwickeln: eine kurze Übersicht 92
3.3 Einrichtung eines Deep-Learning-Rechners 94
3.3.1 Zwei Möglichkeiten, Keras zum Laufen zu bringen 95
3.3.2 Pro und Kontra: Deep Learning in der Cloud ausführen . . . 95
3.3.3 Für Deep Learning geeignete GPUs 96
3.4 Klassifikation von Filmbewertungen: ein Beispiel für eine Binärklassifikation 96
3.4.1 Die IMDb-Datensammlung 96
3.4.2 Daten vorbereiten 98
3.4.3 Neuronales Netz erzeugen 100
3.4.4 Validierung des Ansatzes 104
3.4.5 Vorhersagen über neue Daten mit einem trainierten neuronalen Netz treffen 108
3.4.6 Weitere Experimente 108
3.4.7 Zusammenfassung 109
3.5 Ein Beispiel für eine Mehrfachklassifikation: Klassifikation von Nachrichtenmeldungen 109
3.5.1 Die Reuters-Datensammlung 110
3.5.2 Daten vorbereiten 111
3.5.3 Neuronales Netz erzeugen 112
3.5.4 Validierung des Ansatzes 113
3.5.5 Vorhersagen über neue Daten treffen 116
3.5.6 Eine weitere Möglichkeit zur Handhabung der Klassenbezeichnungen und der Verlustfunktion 116
3.5.7 Hinreichend große zwischenliegende Layer sind wichtig . . . 117
3.5.8 Weitere Experimente 117
3.5.9 Zusammenfassung 118
3.6 Ein Beispiel für eine Regression: Vorhersage der Kaufpreise von Häusern 118
3.6.1 Die Boston-Housing-Price-Datensammlung 118
3.6.2 Daten vorbereiten 119
3.6.3 Neuronales Netz erzeugen 120
3.6.4 K-fache Kreuzvalidierung des Ansatzes 121

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

3.6.5 Zusammenfassung 126
3.7 Zusammenfassung Kapitel 3 126

4 Grundlagen des Machine Learnings 129
4.1 Vier Teilgebiete des Machine Learnings 129
4.1.1 Überwachtes Lernen 129
4.1.2 Unüberwachtes Lernen 130
4.1.3 Selbstüberwachtes Lernen 130
4.1.4 Bestärkendes Lernen 131
4.2 Bewertung von Machine-Learning-Modellen 133
4.2.1 Trainings-, Validierungs- und Testmengen 133
4.2.2 Worauf zu achten ist 137
4.3 Datenvorverarbeitung, Merkmalserstellung und Erlernen von Merkmalen 138
4.3.1 Datenvorverarbeitung für neuronale Netze 138
4.3.2 Merkmalserstellung 140
4.4 Überanpassung und Unteranpassung 142
4.4.1 Das neuronale Netz verkleinern 143
4.4.2 Regularisierung der Gewichte 147
4.4.3 Dropout-Regularisierung 148
4.5 Ein allgemeiner Machine-Learning-Workflow 151
4.5.1 Definition der Aufgabe und Zusammenstellen einer Datenmenge 151
4.5.2 Auswahl eines Erfolgskriteriums 153
4.5.3 Auswahl einer Bewertungsmethode 153
4.5.4 Daten vorbereiten 154
4.5.5 Entwicklung eines Modells, das besser funktioniert als zufälliges Raten 154
4.5.6 Hochskalieren: Entwicklung eines Modells mit Überanpassung 155
4.5.7 Regularisierung des Modells und Abstimmung der Hyperparameter 156
4.6 Zusammenfassung Kapitel 4 157

Teil II Deep Learning in der Praxis 159

5 Deep Learning und maschinelles Sehen 161
5.1 Einführung in CNNs 161
5.1.1 Die Faltungsoperation 165

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

5.1.2 Die Max-Pooling-Operation 171
5.2 Ein CNN von Grund auf mit einer kleinen Datenmenge trainieren 173
5.2.1 Die Bedeutung des Deep Learnings für Aufgaben mit kleinen Datenmengen 174
5.2.2 Daten herunterladen 174
5.2.3 Erstellen des neuronalen Netzes 177
5.2.4 Datenvorverarbeitung 179
5.2.5 Datenaugmentation 182
5.3 Verwendung eines vortrainierten CNNs 188
5.3.1 Merkmalsextraktion 189
5.3.2 Feinabstimmung 200
5.3.3 Zusammenfassung 204
5.4 Visualisierung: Was CNNs erlernen können 205
5.4.1 Visualisierung zwischenliegender Aktivierungen 205
5.4.2 Visualisierung von CNN-Filtern 212
5.4.3 Visualisierung der Heatmaps der Klassenaktivierung 218
5.5 Zusammenfassung Kapitel 5 223

6 Deep Learning, Text und sequenzielle Daten 225
6.1 Textdaten 226
6.1.1 One-hot-Codierung von Wörtern und Zeichen 228
6.1.2 Worteinbettung 231
6.1.3 Zusammengefasst: von reinem Text zu Worteinbettungen 238
6.1.4 Zusammenfassung 245
6.2 Rekurrente neuronale Netze 245
6.2.1 Ein rekurrenter Layer in Keras 248
6.2.2 LSTM- und GRU-Layer 252
6.2.3 Ein konkretes LSTM-Beispiel in Keras 256
6.2.4 Zusammenfassung 258
6.3 Erweiterte Nutzung rekurrenter neuronaler Netze 258
6.3.1 Temperaturvorhersage 258
6.3.2 Daten vorbereiten 261
6.3.3 Eine vernünftige Abschätzung ohne Machine Learning 266
6.3.4 Ein elementarer Machine-Learning-Ansatz 267
6.3.5 Ein erstes RNN 269
6.3.6 Rekurrentes Dropout-Verfahren zum Verhindern einer Überanpassung 271

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

6.3.7 Hintereinanderschaltung rekurrenter Layer . . . 273
6.3.8 Bidirektionale RNNs . . . 275
6.3.9 Noch einen Schritt weiter gehen . . . 280
6.3.10 Zusammenfassung . . . 280
6.4 Verarbeitung von Sequenzen mit CNNs . . . 282
6.4.1 Eindimensionale Faltung sequenzieller Daten . . . 282
6.4.2 Eindimensionales Pooling sequenzieller Daten . . . 283
6.4.3 Implementierung eines eindimensionalen CNNs . . . 283
6.4.4 Lange Sequenzen mit einer Kombination aus CNNs und RNNs verarbeiten . . . 286
6.4.5 Zusammenfassung . . . 290
6.5 Zusammenfassung Kapitel 6 . . . 291

7 Bewährte Verfahren des Deep Learnings . . . 293
7.1 Jenseits des Sequential-Modells: die funktionale Keras-API . . . 293
7.1.1 Einführung in die funktionale API . . . 297
7.1.2 Modelle mit mehreren Eingaben . . . 299
7.1.3 Modelle mit mehreren Ausgaben . . . 302
7.1.4 Gerichtete azyklische Graphen von Layern . . . 306
7.1.5 Gemeinsam genutzte Gewichte von Layern . . . 311
7.1.6 Modelle als Layer . . . 312
7.1.7 Zusammenfassung . . . 314
7.2 Deep-Learning-Modelle mit Callbacks und TensorBoard untersuchen und überwachen . . . 314
7.2.1 Beeinflussung eines Modells während des Trainings durch Callbacks . . . 314
7.2.2 Einführung in das Visualisierungs-Framework TensorBoard . . . 319
7.2.3 Zusammenfassung . . . 325
7.3 Modelle richtig ausreizen . . . 325
7.3.1 Erweiterte Architekturmuster . . . 325
7.3.2 Hyperparameteroptimierung . . . 329
7.3.3 Ensemblemodelle . . . 331
7.3.4 Zusammenfassung . . . 334
7.4 Zusammenfassung Kapitel 7 . . . 335

8 Generatives Deep Learning . . . 337
8.1 Texterzeugung mit LSTM-Modellen . . . 339
8.1.1 Eine kurze Geschichte generativer RNNs . . . 339

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by
mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

8.1.2 Wie erzeugt man sequenzielle Daten? 340
8.1.3 Die Bedeutung der Sampling-Strategie 341
8.1.4 Implementierung der LSTM-Texterzeugung für Zeichen . . . 343
8.1.5 Zusammenfassung 350
8.2 DeepDream 350
8.2.1 DeepDream in Keras implementieren 351
8.2.2 Zusammenfassung 359
8.3 Stilübertragung mit dem Neural-Style-Algorithmus 359
8.3.1 Verlustfunktion für den Inhalt 360
8.3.2 Verlustfunktion für den Stil 361
8.3.3 Stilübertragung in Keras 362
8.3.4 Zusammenfassung 370
8.4 Bilderzeugung mit Variational Autoencoders 370
8.4.1 Sampling eines latenten Bilderraums 371
8.4.2 Konzeptvektoren für das Bearbeiten von Bildern 372
8.4.3 Variational Autoencoders 372
8.4.4 Zusammenfassung 380
8.5 Einführung in Generative-Adversarial-Netze 381
8.5.1 Eine schematische GAN-Implementierung 383
8.5.2 Einige nützliche Tricks 384
8.5.3 Der Generator 385
8.5.4 Der Diskriminator 386
8.5.5 Das gegnerische Netz 387
8.5.6 Training des DCGAN 388
8.5.7 Zusammenfassung 391
8.6 Zusammenfassung Kapitel 8 392

9 Schlussfolgerungen 393
9.1 Kernkonzepte im Überblick 393
9.1.1 Verschiedene Ansätze der KI 394
9.1.2 Die Besonderheiten des Deep Learnings 394
9.1.3 Was ist vom Deep Learning zu halten? 395
9.1.4 Wichtige zugrunde liegende Technologien 397
9.1.5 Der allgemeine Machine-Learning-Workflow 397
9.1.6 Wichtige Netzarchitekturen 398
9.1.7 Der Raum der Möglichkeiten 404
9.2 Grenzen des Deep Learnings 406
9.2.1 Das Risiko der Vermenschlichung von Deep-Learning-Modellen 406

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by
mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

9.2.2 Lokale und extreme Generalisierung 409
9.2.3 Zusammenfassung 411
9.3 Die Zukunft des Deep Learnings 411
9.3.1 Modelle als Programme 412
9.3.2 Jenseits von Backpropagation und differenzierbaren Layern 414
9.3.3 Automatisiertes Machine Learning 415
9.3.4 Beständiges Lernen und Wiederverwendung modularer Subroutinen 416
9.3.5 Langfristige Aussichten 418
9.4 Auf dem Laufendem bleiben 419
9.4.1 Praktische Erfahrungen sammeln mit Kaggle 419
9.4.2 Aktuelle Entwicklungen auf der arXiv-Website nachlesen 420
9.4.3 Erkundung des Keras-Ökosystems 420
9.5 Schlusswort 421

A Installation von Keras und der Erweiterungen unter Ubuntu 423
A.1 Überblick über den Installationsvorgang 423
A.2 Installation der erforderlichen Erweiterungen 423
A.3 Einrichtung der GPU-Unterstützung 424
A.3.1 CUDA installieren 424
A.3.2 cuDNN installieren 425
A.3.3 Die CUDA-Umgebung 425
A.4 Keras und TensorFlow installieren 427

B RStudio-Server auf einer EC2-GPU-Instanz betreiben 429
B.1 Gründe, Deep Learning auf AWS zu betreiben 429
B.2 Gründe, auf AWS zu verzichten 429
B.3 Einrichtung einer AWS-GPU-Instanz 430
B.3.1 R und RStudio installieren 432
B.3.2 CUDA konfigurieren 433
B.3.3 Voraussetzungen für Keras 434
B.4 Auf den RStudio-Server zugreifen 434
B.5 Keras installieren 436

Stichwortverzeichnis 437

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

Vorwort

Da Sie dieses Buch lesen, ist Ihnen vermutlich bewusst, welchen außerordentlichen Fortschritt Deep Learning auf dem Gebiet der KI (der Künstlichen Intelligenz) in jüngster Zeit darstellt. In nur fünf Jahren hat sich die Bild- und Spracherkennung von einem nahezu unbrauchbaren Zustand zu einem Werkzeug mit fast übermenschlichen Fähigkeiten entwickelt.

Die mit diesem Fortschritt einhergehenden Folgen betreffen fast sämtliche Industriezweige. Für die Anwendung von Deep-Learning-Technologien auf alle erdenklichen Aufgaben müssen wir sie so vielen Menschen wie möglich zugänglich machen – auch Laien, die keine Forscher oder Doktoranden sind. Deep Learning muss radikal demokratisiert werden, damit es sein volles Potenzial ausschöpfen kann.

Als ich im März 2015 die erste Version des Deep-Learning-Frameworks Keras veröffentlichte, hatte ich eigentlich die Demokratisierung der KI gar nicht im Sinn. Ich hatte mehrere Jahre auf dem Gebiet des Machine Learnings geforscht und Keras für meine eigenen Experimente entwickelt. Aber in den Jahren 2015 und 2016 befassten sich immer mehr Menschen mit Deep Learning – Zehntausende. Viele davon entschieden sich für Keras, weil es das für Einsteiger am einfachsten verwendbare Framework war und noch immer ist. Nachdem ich beobachten konnte, dass Scharen von Neulingen Keras auf unerwartete, leistungsstarke Weise einsetzten, wurde mir klar, dass ich mich um die Zugänglichkeit und Demokratisierung der KI bemühen sollte. Ich stellte fest, dass diese Technologien umso nützlicher und wertvoller wurden, je weiter sie sich verbreiteten. Bei der Entwicklung von Keras wurde die einfache Zugänglichkeit schnell zu einem ausdrücklichen Ziel, und die Entwicklergemeinde erzielte in einigen wenigen Jahren bemerkenswerte Erfolge auf diesem Gebiet. Wir ermöglichten Zehntausenden von Entwicklern, Deep Learning einzusetzen, die es ihrerseits zum Lösen bedeutsamer Probleme verwendeten, von denen wir bis vor Kurzem gar nicht wussten, dass es sie überhaupt gibt.

Das Buch, das Sie in Ihren Händen halten, ist ein weiterer Schritt auf dem Weg, Deep Learning für so viele Menschen wie möglich verfügbar zu machen. Es war schon lange überfällig, einen begleitenden Kurs für Keras zu erarbeiten, der nicht nur die Grundlagen des Deep Learnings behandelt, sondern gleichzeitig auch typische Anwendungsmuster und bewährte Verfahren in Keras aufzeigt. Dieses Buch stellt meinen Versuch dar, solch einen Lehrgang anzubieten. Ich habe mich beim

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by
mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

Schreiben darauf konzentriert, die dem Deep Learning zugrunde liegenden Konzepte und deren Implementierung so verständlich wie möglich zu schildern. Dabei habe ich keinerlei Vereinfachungen vorgenommen – ich bin der festen Überzeugung, dass es beim Deep Learning keine wirklich schwierigen Konzepte gibt. Ich hoffe, dass Ihnen dieses Buch von Nutzen sein wird und Ihnen dabei hilft, intelligente Anwendungen für die Aufgaben zu entwickeln, die für Sie von Bedeutung sind.

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by
mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

Einleitung

Deep Learning mit R richtet sich an Statistiker, Analytiker, Entwickler und Studenten, die bereits Erfahrung mit der Programmiersprache R gesammelt haben, aber noch keine nennenswerten Kenntnisse über Machine Learning und Deep Learning besitzen. Das Buch ist eine Überarbeitung des zuvor erschienenen Titels *Deep Learning mit Python* (mitp Verlag, 2018) und verwendet für alle Codebeispiele die Keras-Schnittstelle in R. Ziel des Buchs ist es, der R-Community eine Ressource zu bieten, die sowohl die grundlegende Theorie als auch fortgeschrittene praktische Anwendungen behandelt. Sie werden anhand von mehr als 30 Codebeispielen mit ausführlichen Kommentaren, praxisnahen Empfehlungen und zusammenfassenden Erklärungen alles erfahren, was Sie benötigen, um Deep Learning zum Lösen konkreter Aufgabenstellungen einzusetzen.

Die Codebeispiele verwenden das Deep-Learning-Framework Keras, das auf TensorFlow aufbaut. Keras ist eines der beliebtesten und am schnellsten wachsenden Deep-Learning-Frameworks und empfiehlt sich als bestens geeignetes Tool zum Einstieg ins Deep Learning. Nach der Lektüre dieses Buchs werden Sie über solide Kenntnisse des Deep Learnings verfügen und beurteilen können, wann es anwendbar ist und wo es an Grenzen stößt. Sie werden mit dem typischen Workflow bei der Bewältigung von für das Machine Learning geeigneten Aufgaben vertraut sein und wissen, wie sich häufig auftretende Probleme lösen lassen. Sie werden in der Lage sein, praxisnahe Aufgaben vom maschinellen Sehen bis zur Sprachverarbeitung mit Keras anzugehen: Bildklassifikation, Vorhersage von Zeitreihen, Stimmungsanalyse, Erzeugung von Bildern und Texten und vieles andere mehr.

Wer sollte dieses Buch lesen?

Das Buch wendet sich an Leser mit Erfahrung in der Programmierung mit R, die ins Machine Learning und Deep Learning einsteigen möchten. Es kann jedoch auch für viele andere Leser von Nutzen sein:

- Falls Sie Data Scientist sind und Ihnen Machine Learning bereits vertraut ist, bietet das Buch eine solide und praxisnahe Einführung ins Deep Learning, das am schnellsten wachsende und bedeutendste Teilgebiet des Machine Learnings.

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by
mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

- Falls Sie Deep-Learning-Experte sind und sich mit dem Keras-Framework vertraut machen möchten, werden Sie feststellen, dass dieses Buch den besten verfügbaren Keras-Schnellkurs darstellt.
- Falls Sie Doktorand sind und sich in einer formalen Umgebung mit Deep Learning befassen, werden Sie feststellen, dass dieses Buch eine praxisorientierte Ergänzung Ihrer Ausbildung ist, die Ihnen eine Vorstellung vom Verhalten tiefer neuronaler Netze vermittelt und Sie mit den wichtigsten bewährten Verfahren vertraut macht.

Auch technisch Interessierte, die selbst nicht regelmäßig programmieren, werden das Buch als Einführung in die grundlegenden und weiterführenden Konzepte des Deep Learnings nützlich finden. Für den Einsatz von Keras sind hinreichende R-Kenntnisse unverzichtbar. Erfahrung mit Machine Learning oder Deep Learning wird nicht vorausgesetzt: Das Buch erläutert alle erforderlichen Grundlagen. Fortgeschrittene Mathematikkenntnisse sind ebenfalls nicht notwendig – Kenntnisse auf Oberschulniveau sollten für das Verständnis ausreichen.

Überblick

Das Buch besteht aus zwei Teilen. Sofern Sie noch keine Erfahrung mit Machine Learning haben, sollten Sie zunächst Teil I lesen, bevor Sie sich mit Teil II befassen. Wir werden anfangs einfache Beispiele behandeln und uns im weiteren Verlauf des Buchs zunehmend Verfahren annähern, die dem Stand der Technik entsprechen.

Teil I stellt eine allgemeine Einführung in das Deep Learning dar, die Zusammenhänge aufzeigt und Begriffe definiert sowie alle für den Einstieg ins Deep Learning und neuronale Netze erforderlichen Konzepte erklärt:

- In Kapitel 1 wird das wesentliche Hintergrundwissen rund um die Themen KI, Machine Learning und Deep Learning erläutert.
- Kapitel 2 stellt die für das Verständnis von Deep Learning erforderlichen grundlegenden Konzepte vor: Tensoren, Tensoroperationen sowie Gradientenabstiegs- und Backpropagation-Verfahren. Dieses Kapitel enthält auch das erste Beispiel für ein funktionierendes neuronales Netz.
- Kapitel 3 enthält alles, was Sie wissen müssen, um neuronale Netze zu verwenden: eine Einführung in Keras, das Deep-Learning-Framework der Wahl, einen Leitfaden zum Einrichten Ihres Arbeitsplatzrechners und drei grundlegende Codebeispiele mit ausführlichen Erklärungen. Am Ende dieses Kapitels werden Sie in der Lage sein, einfache neuronale Netze für Klassifikations- und Regressionsaufgaben zu trainieren. Außerdem werden Sie eine klare Vorstellung davon haben, was beim Trainieren eigentlich im Hintergrund vor sich geht.

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by
mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

- Kapitel 4 erörtert den typischen Workflow beim Machine Learning. Darüber hinaus lernen Sie einige häufig auftauchende Fallstricke kennen und erfahren, wie man diese umgeht.

Teil II befasst sich ausführlich mit den praktischen Anwendungen des Deep Learnings beim maschinellen Sehen (engl. *Computer Vision*) und bei der Verarbeitung natürlicher Sprache. Viele der hier vorgestellten Beispiele können Ihnen als Vorlage zum Lösen von Problemen dienen, die Ihnen in der Praxis des Deep Learnings begegnen werden:

- Kapitel 5 untersucht eine Reihe von Beispielen für das maschinelle Sehen und legt dabei einen Schwerpunkt auf die Klassifikation von Bildern.
- In Kapitel 6 sammeln Sie Erfahrung mit Verfahren zur Verarbeitung sequenzieller Daten, wie etwa Texten und Zeitreihen.
- Kapitel 7 stellt einige erweiterte Verfahren zum Erstellen von Deep-Learning-Modellen vor, die dem gegenwärtigen Stand der Technik entsprechen.
- Kapitel 8 erläutert sogenannte generative Modelle: Deep-Learning-Modelle, die in der Lage sind, Bilder und Texte zu erzeugen, und manchmal erstaunlich künstlerisch wirkende Ergebnisse erzielen.
- Kapitel 9 fasst die im Buch erörterten Inhalte zusammen, wirft einen Blick auf die Beschränkungen des Deep Learnings und erkundet die voraussichtliche Zukunft dieses Fachgebiets.

Erforderliche Hard- und Software

Sämtliche Codebeispiele in diesem Buch verwenden das quelloffene und kostenlose Deep-Learning-Framework Keras (`https://keras.rstudio.com`). Sie sollten über Zugang zu einem UNIX-System verfügen. Es ist zwar auch möglich, Windows zu verwenden, davon raten wir jedoch ab.

Darüber hinaus empfehlen wir, eine neuere NVIDIA-GPU wie z. B. die TITAN X zu verwenden. Das ist zwar nicht unbedingt notwendig, erleichtert die Arbeit aber ungemein, da die Codebeispiele erheblich schneller ausgeführt werden. Weitere Informationen zur Einrichtung eines Arbeitsplatzrechners mit GPU finden Sie unter `https://tensorflow.rstudio.com/tools/local_gpu`.

Sollte Ihnen kein Zugang zu einem Arbeitsplatzrechner mit einer neueren NVIDIA-GPU zur Verfügung stehen, können Sie stattdessen eine Cloud-Umgebung (wie etwa eine n1-standard-8-Instanz mit zusätzlicher NVIDIA Tesla K80) oder von Amazon Web Services (AWS) bereitgestellte GPU-Instanzen (wie z. B. eine p2.xlarge-Instanz) verwenden. Weitere Informationen über die verfügbaren Möglichkeiten finden Sie unter `https://tensorflow.rstudio.com/tools/cloud_gpu`.

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by
mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

Quellcode

Sämtliche Codebeispiele stehen in Form von R-Notebooks zum Herunterladen bereit (unter `https://github.com/jjallaire/deep-learning-with-r-notebooks`).

Das Forum zum Buch

Der amerikanische Originalverlag Manning Publications betreibt unter `https://forums.manning.com/forums/deep-learning-with-r` ein Webforum zum Buch. Dort können Sie Kommentare schreiben, technische Fragen stellen und Hilfe von den Autoren und anderen Forumsteilnehmern erhalten. Weitere Information und Hinweise zu den Verhaltensregeln finden Sie unter `https://forums.manning.com/forums/about`.

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by
mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

Danksagungen

Ich danke der Keras-Community dafür, dass sie dieses Buch ermöglicht hat. Inzwischen gibt es mehrere Hundert Entwickler, die Beiträge leisten, und mehr als 200.000 Benutzer. Eure Beiträge und euer Feedback haben Keras zu dem gemacht, was es heute ist.

Außerdem danke ich Google für die Unterstützung des Keras-Projekts. Es war großartig zu erleben, dass Keras als TensorFlows High-Level-API übernommen wurde. Die reibungslose Zusammenführung von Keras und TensorFlow ist sowohl für TensorFlow- als auch für Keras-User von Vorteil und ermöglicht fast allen Entwicklern den Zugang zum Deep Learning.

Ich danke den Mitarbeitern des amerikanischen Originalverlags Manning, die dieses Buch ermöglicht haben: dem Herausgeber Marjan Bace und allen Redaktions- und Produktionsmitgliedern, insbesondere Jennifer Stout, Janet Vail, Tiffany Taylor, Katie Tennant, Dottie Marsico und vielen anderen, die hinter den Kulissen tätig waren.

Großer Dank gebührt auch dem von Aleksandar Dragosavljevic geleiteten technischen Korrektorenteam – Diego Acuña Rozas, Geoff Barto, David Blumenthal-Barby, Abel Brown, Clark Dorman, Clark Gaylord, Thomas Heiman, Wilson Mar, Sumit Pal, Vladimir Pasman, Gustavo Patino, Peter Rabinovitch, Alvin Raj, Claudio Rodriguez, Srdjan Santic, Richard Tobias, Martin Verzilli, William E. Wheeler und Daniel Williams – sowie den Mitgliedern des Forums. Sie wiesen auf technische Fehler, unpassende Terminologie und Tippfehler hin und lieferten Themenvorschläge. Alle diese Überprüfungen und die Rückmeldungen zu den im Forum erörterten Themen haben das Manuskript mitgeformt und mitgestaltet.

Was technische Fragen betrifft, gebührt Jerry Gaines, der als technischer Redakteur tätig war, sowie den technischen Korrekturlesern Alex Ott und Richard Tobias besonderer Dank. Bessere technische Redakteure hätte ich mir nicht wünschen können.

Und abschließend möchte ich meiner Frau Maria meinen Dank für die unglaubliche Unterstützung während der Entwicklung von Keras und beim Schreiben dieses Buchs aussprechen.

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by
mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

Über die Autoren

François Chollet ist bei Google in Mountain View, Kalifornien, tätig und befasst sich mit Deep Learning. Er ist der Entwickler der Deep-Learning-Bibliothek Keras und hat bedeutende Beiträge zum Machine-Learning-Framework TensorFlow geleistet. Er forscht auf dem Gebiet des Deep Learnings mit den Schwerpunkten maschinelles Sehen und der Anwendung des Machine Learnings auf formales Schließen. Seine Forschungsergebnisse wurden auf bedeutenden Veranstaltungen des Fachgebiets veröffentlicht, unter anderem auf der *Conference on Computer Vision and Pattern Recognition* (CVPR), der *Conference on Neural Information Processing Systems* (NIPS), der *International Conference on Learning Representations* (ICLR) und weiteren.

J.J. Allaire ist der Begründer von RStudio und der Entwickler der RStudio-IDE. Allaire ist außerdem der Entwickler der TensorFlow- und Keras-Schnittstellen in R.

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by
mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

Kapitel 1

Was ist Deep Learning?

Die Themen in diesem Kapitel:

- Allgemeine Definitionen und grundlegende Konzepte
- Chronologie der Entwicklung des Machine Learnings
- Entscheidende Faktoren bei der zunehmenden Verbreitung des Deep Learnings und zukünftiges Potenzial

In den vergangenen Jahren sorgte die Künstliche Intelligenz (engl. *Artificial Intelligence* oder kurz AI) für einen weitreichenden Medienrummel. Machine Learning, Deep Learning und KI waren Gegenstand unzähliger Artikel, häufig auch jenseits technologieorientierter Publikationen. Man hat uns eine Zukunft mit intelligenten Chatbots, selbstfahrenden Autos und virtuellen Assistenten versprochen – eine Zukunft, die mitunter düster beschrieben wird, aber auch als eine Utopie, in der menschliche Arbeit selten ist und die meisten wirtschaftlichen Tätigkeiten von Robotern oder KI-Agenten erledigt werden. Für zukünftige und heutige Anwender des Machine Learnings ist es von großer Bedeutung, das Signal in all dem Rauschen zu erkennen, um wirklich weltbewegende Entwicklungen von hochgejubelten Pressemitteilungen unterscheiden zu können. Hier steht nicht weniger als unsere Zukunft auf dem Spiel – eine Zukunft, in der Sie eine aktive Rolle einnehmen müssen: Nach der Lektüre dieses Buchs gehören Sie zu denjenigen, die KI-Agenten entwickeln werden. Setzen wir uns also mit den folgenden Fragen auseinander: Was kann Deep Learning heute schon leisten? Wie bedeutsam ist es? Worauf steuern wir zu? Darf man dem Hype Glauben schenken? Dieses Kapitel erläutert das wesentliche Hintergrundwissen rund um die Themen KI, Machine Learning und Deep Learning.

1.1 Künstliche Intelligenz, Machine Learning und Deep Learning

Zunächst einmal müssen wir eindeutig definieren, was wir eigentlich meinen, wenn es um KI geht. Was sind KI, Machine Learning und Deep Learning (siehe Abbildung 1.1) eigentlich genau? In welcher Beziehung stehen sie zueinander?

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by
mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

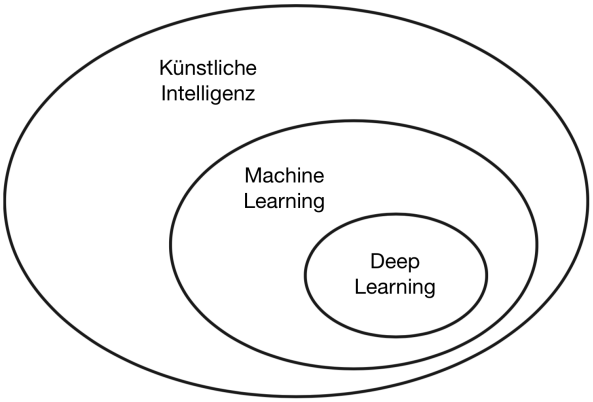

Abb. 1.1: Künstliche Intelligenz, Machine Learning und Deep Learning

1.1.1 Künstliche Intelligenz

Die Künstliche Intelligenz wurde in den 1950er-Jahren entwickelt, als sich einige Pioniere der damals aufblühenden Informatik fragten, ob es möglich sei, einem Computer das »Denken« beizubringen – eine Frage, mit deren Auswirkungen wir uns auch heute noch befassen. Eine kompakte Definition dieses Fachgebiets lautet folgendermaßen: *der Versuch, normalerweise von Menschen erledigte geistige Aufgaben automatisiert zu lösen*. In diesem Sinne ist die KI ein allgemeines Fachgebiet, das Machine Learning und Deep Learning einschließt, aber auch viele andere Ansätze umfasst, die nichts mit Lernen zu tun haben. So verwendeten beispielsweise die ersten Schachprogramme lediglich einen von den Programmierern fest vorgegebenen Regelsatz – dabei handelte es sich jedoch nicht um Machine Learning. Viele Experten gingen lange davon aus, dass sich eine KI auf menschlichem Niveau durch einen hinreichend großen Regelsatz zur Verarbeitung von Wissen erreichen ließe. Dieser Ansatz ist unter der Bezeichnung *symbolische KI* bekannt und war von Mitte der 1950er- bis Ende der 1980er-Jahre das vorherrschende Paradigma der KI. Diese Sichtweise erreichte Mitte der 1980er-Jahre während des Booms der sogenannten *Expertensysteme* ihren Höhepunkt.

Die symbolische KI erwies sich zwar als durchaus brauchbar, um wohldefinierte logische Aufgaben wie etwa Schachspielen zu lösen, es gelang jedoch nicht, explizite Regeln zur Lösung komplexer, weniger deutlich umrissener Aufgabenstellungen zu finden, wie z. B. die Klassifikation von Bildern, die Erkennung natürlicher Sprache und die Übersetzung von Fremdsprachen. So ergab sich ein neuer Ansatz, der den Platz der symbolischen KI einnehmen sollte: *Machine Learning*.

1.1.2 Machine Learning

Lady Ada Lovelace war im viktorianischen England mit Charles Babbage, dem Erfinder der *Analytical Engine* (engl. für *analytische Maschine*), befreundet und arbei-

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by
mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

tete mit ihm zusammen. Die Analytical Engine war visionär und ihrer Zeit weit voraus, allerdings war sie nicht als Allzweckcomputer gedacht, als sie während der 1830er- und 1840er-Jahre entworfen wurde, denn das Konzept eines Allzweckcomputers musste erst noch erfunden werden. Sie war lediglich dafür ausgelegt, bestimmte Berechnungen auf dem Fachgebiet der mathematischen Analyse durch mechanische Vorgänge zu automatisieren – daher auch der Name Analytical Engine. Ada Lovelace kommentierte die Erfindung 1843 folgendermaßen: »Die Analytical Engine beansprucht für sich in keinerlei Weise, Neues zu erschaffen. Sie kann das leisten, von dem wir wissen, wie wir es befehlen können ... Sie dient dazu, uns dabei zu unterstützen, uns bereits Bekanntes bereitzustellen.«

Diese Anmerkung wurde 1950 von Alan Turing in seiner bahnbrechenden Arbeit *Computing Machinery and Intelligence*[1] zitiert und als »Lady Lovelaces Einspruch« bezeichnet. In dieser Arbeit schlug er den Turing-Test sowie weitere entscheidende die KI prägende Konzepte vor. Turing zitierte Ada Lovelace, während er darüber nachgrübelte, ob Allzweckcomputer in der Lage wären, zu lernen oder originell zu sein. Er kam zu dem Schluss, dass dies der Fall sei.

Machine Learning entstand aufgrund folgender Frage: Könnte ein Computer über das, »von dem wir wissen, wie wir es befehlen können«, hinausgehen und selbst erlernen, wie eine bestimmte Aufgabe erledigt wird? Könnte ein Computer uns überraschen? Könnte ein Computer automatisch Regeln erlernen, indem er Daten betrachtet, ohne dass Programmierer diese Datenverarbeitungsregeln von Hand erstellen müssen?

Diese Fragen öffneten einem neuen Programmierparadigma Tür und Tor. Bei der klassischen Programmierung, der symbolischen KI, geben Menschen Regeln (ein Programm) und die gemäß diesen Regeln zu verarbeitenden Daten vor, was zu Antworten führt (siehe Abbildung 1.2). Beim Machine Learning geben Menschen sowohl die Daten als auch die dazugehörigen Antworten vor, und heraus kommen die Regeln. Diese Regeln sind dann auf neue Daten anwendbar und liefern eigenständige Antworten.

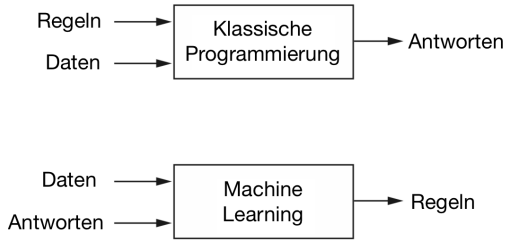

Abb. 1.2: Machine Learning, ein neues Programmierparadigma

1 A. M. Turing, *Computing Machinery and Intelligence*, Mind 59, Nr. 236 (1950), Seiten 433–460.

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by
mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

Ein Machine-Learning-System wird also nicht explizit programmiert, sondern vielmehr *trainiert*. Dem System werden viele für die zu lösende Aufgabe relevante Beispiele bereitgestellt, in denen es nach einer statistischen Struktur sucht, die ihm letztendlich erlaubt, Regeln für die Automatisierung der Aufgabe zu erstellen. Wenn Sie beispielsweise die Verschlagwortung Ihrer Urlaubsfotos automatisieren möchten, könnten Sie dem System Ihre bereits von Menschen verschlagworteten Bilder zur Verfügung stellen. Das System würde dann statistische Regeln erlernen, um bestimmten Fotos bestimmte Schlagwörter zuzuweisen.

So richtig blühte das Machine Learning zwar erst in den 1990er-Jahren auf, es wurde jedoch dank der Verfügbarkeit schnellerer Hardware und größerer Datenmengen rasch zum verbreitetsten und erfolgreichsten Teilgebiet der KI. Machine Learning ist eng mit der mathematischen Statistik verwandt, unterscheidet sich aber in einigen wichtigen Punkten. Im Gegensatz zur Statistik kommen beim Machine Learning tendenziell sehr große, komplexe Datenmengen zum Einsatz (wie z. B. eine Datenmenge, die aus mehreren Millionen Fotos mit jeweils Zehntausenden von Pixeln besteht), für die klassische statistische Verfahren wie eine Bayes'sche Analyse nicht praktikabel wären. Daher spielt die mathematische Theorie beim Machine Learning und insbesondere beim Deep Learning nur eine vergleichsweise kleine – vielleicht zu kleine – Rolle. In diesem praxisorientierten Fachgebiet werden Ideen häufiger empirisch erprobt als theoretisch vorhergesagt.

1.1.3 Die Repräsentation anhand der Daten erlernen

Um Deep Learning zu definieren und um den Unterschied zwischen Deep Learning und anderen Ansätzen des Machine Learnings zu verstehen, müssen wir zunächst einmal eine Vorstellung davon erlangen, wie Machine-Learning-Algorithmen eigentlich funktionieren. Ich habe soeben dargelegt, dass beim Machine Learning anhand von Beispielen für die zu erwartenden Ergebnisse Regeln gesucht werden, um die Verarbeitung von Daten zu erledigen. Für das Machine Learning sind also drei Dinge erforderlich:

- *Eingabedaten* – Bei einer Spracherkennung könnte es sich bei den Eingabedaten beispielsweise um Tondateien handeln, die Sprachaufnahmen enthalten, oder bei der Verschlagwortung von Fotos um Bilddateien.
- *Beispiele für die zu erwartende Ausgabe* – Bei einer Spracherkennung könnten die Beispiele in Form von durch Menschen erstellten Textdateien vorliegen, die den Inhalt der Tondateien wiedergeben. Bei einer Bilderkennung wären die zu erwartenden Ausgaben Kennzeichnungen wie »Hund«, »Katze« usw.
- *Eine Möglichkeit, zu messen, ob der Algorithmus gut funktioniert* – Diese Messung ist erforderlich, um die Abweichungen der aktuellen Ausgabe des Algorithmus von der zu erwartenden Ausgabe zu ermitteln. Das Messergebnis dient als Feedback-Signal zur Anpassung der Funktionsweise des Algorithmus. Diese Anpassung ist das, was wir als *Lernen* bezeichnen.

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by
mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

Ein Machine-Learning-Modell wandelt die Eingabedaten in sinnvolle Ausgaben um. Dieser Vorgang wird anhand der bekannten Beispiele für Ein- und Ausgaben »erlernt«. Die grundsätzliche Aufgabe beim Machine Learning und beim Deep Learning besteht darin, *Daten sinnvoll umzuwandeln*. Mit anderen Worten: Es müssen sinnvolle *Repräsentationen* der gegebenen Eingabedaten erlernt werden – Repräsentationen, die uns der zu erwartenden Ausgabe näherbringen. Aber bevor wir fortfahren: Was genau ist eine Repräsentation? Im Grunde genommen handelt es sich um eine andere Art, Daten zu betrachten – Daten zu *repräsentieren* oder zu *codieren*. Ein Farbfoto kann beispielsweise im RGB-Format (Rot, Grün, Blau) oder im HSV-Format (*Hue, Saturation, Value*, Farbwert, Farbsättigung und Helligkeitswert) codiert sein. Dabei handelt es sich um zwei unterschiedliche Repräsentationen derselben Daten. Manche Aufgaben, die mit einer der Repräsentationen schwierig sind, können in einer anderen Repräsentation ganz einfach sein. Die Aufgabe, alle roten Pixel eines Bilds auszuwählen, ist im RGB-Format einfacher, die Aufgabe, die Farbsättigung zu verringern, ist hingegen im HSV-Format einfacher. Bei Machine-Learning-Modellen geht es immer auch darum, angemessene Repräsentationen der Eingabedaten zu finden – Transformationen der Daten, die eine gegebene Aufgabe vereinfachen, wie z. B. eine Klassifikation.

Betrachten wir ein konkretes Beispiel, nämlich ein aus einer x- und einer y-Achse bestehendes Koordinatensystem sowie einige Punkte, die durch ihre `(x,y)`-Koordinaten definiert sind (siehe Abbildung 1.3).

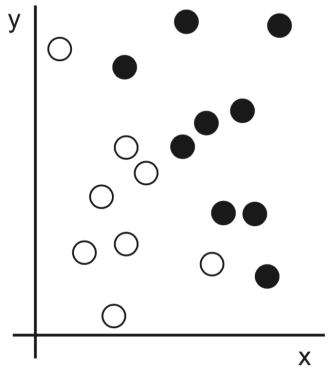

Abb. 1.3: Einige Beispieldaten

Wie Sie sehen, gibt es ein paar weiße und einige schwarze Punkte. Nehmen wir an, wir möchten einen Algorithmus entwickeln, der die `(x,y)`-Koordinaten eines Punkts entgegennimmt und ausgibt, ob der Punkt wahrscheinlich schwarz oder weiß ist. Hier gilt:

- Die Eingaben sind die Koordinaten der Punkte.
- Die zu erwartenden Ausgaben sind die Farben der Punkte.

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by
mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

- Eine Möglichkeit, zu messen, ob der Algorithmus gut funktioniert, wäre beispielsweise der Prozentsatz der richtig klassifizierten Punkte.

Wir benötigen hier eine neue Repräsentation der Daten, die weiße und schwarze Punkte eindeutig voneinander trennt. Eine von vielen möglichen Transformationen wäre beispielsweise, wie in Abbildung 1.4 gezeigt, ein Wechsel des Koordinatensystems.

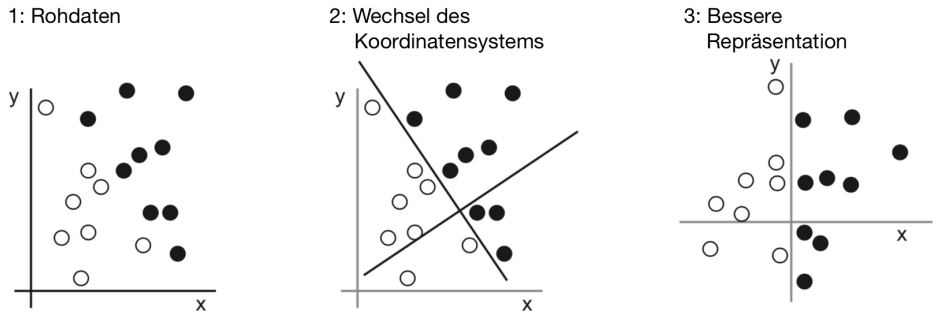

Abb. 1.4: Wechsel des Koordinatensystems

In diesem neuen Koordinatensystem stellen die Koordinaten der Punkte eine andere Repräsentation der Daten dar. Und zwar eine bessere! Bei dieser Repräsentation kann die Klassifikation der Punkte als schwarz oder weiß durch eine einfache Regel formuliert werden: »Für schwarze Punkte gilt $x > 0$« oder »Für weiße Punkte gilt $x < 0$.« Diese neue Repräsentation löst die Klassifikationsaufgabe schon.

In diesem Fall haben wir das Koordinatensystem von Hand neu definiert. Wenn wir stattdessen systematisch nach möglichen Änderungen des Koordinatensystems suchen und den Prozentsatz der korrekt klassifizierten Punkte als Feedback verwenden, betreiben wir Machine Learning. Im Kontext des Machine Learnings beschreibt das *Learning* die automatische Suche nach besseren Repräsentationen.

Machine-Learning-Algorithmen bestehen stets aus der automatischen Suche nach solchen Transformationen, die für eine gegebene Aufgabe nützlichere Repräsentationen der Daten liefern. Bei diesen Operationen kann es sich, wie Sie gerade gesehen haben, um den Wechsel des Koordinatensystems, lineare Projektionen (bei denen Informationen verloren gehen können), Parallelverschiebungen, nicht lineare Operationen (wie etwa »Wähle alle Punkte aus, für die $x > 0$ gilt«) usw. handeln. Machine-Learning-Algorithmen sind bei der Suche nach solchen Transformationen für gewöhnlich nicht sonderlich kreativ, sondern durchsuchen einfach nur eine vorgegebene Menge von Operationen, die als *Hypothesenraum* bezeichnet wird.

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by
mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

Technisch betrachtet ist Machine Learning also die Suche nach nützlichen Repräsentationen der Eingabedaten in einer vorgegebenen Menge von Möglichkeiten unter Berücksichtigung eines Feedback-Signals. Diese einfache Idee ermöglicht es, geistige Aufgaben von bemerkenswerter Bandbreite zu lösen, die von der Spracherkennung bis zu selbstfahrenden Autos reichen.

Nachdem wir nun geklärt haben, was mit *Learning* gemeint ist, wenden wir uns der Frage nach dem Besonderen des *Deep Learnings* zu.

1.1.4 Das »Deep« in Deep Learning

Deep Learning ist ein Teilgebiet des Machine Learnings: ein neuer Ansatz, die Repräsentationen anhand von Daten zu erkennen, der den Schwerpunkt auf das Erlernen aufeinanderfolgender *Layer* (Schichten) mit zunehmend aussagekräftigeren Repräsentationen legt. Das *Deep* in Deep Learning bezieht sich also nicht auf irgendein tiefer gehendes durch diesen Ansatz erzielbares Verständnis, sondern steht für das Konzept aufeinanderfolgender Repräsentations-Layer. Die Anzahl der zu einem Datenmodell beitragenden Layer wird als die *Tiefe* des Modells bezeichnet. Man hätte Deep Learning auch als *Lernen durch schichtweise Repräsentationen* oder *Lernen durch hierarchische Repräsentationen* bezeichnen können. Deep Learning umfasst heutzutage oft Dutzende oder sogar Hunderte aufeinanderfolgender Repräsentations-Layer – die alle durch die Bereitstellung der Trainingsdaten automatisch erlernt werden. Andere Ansätze des Machine Learnings konzentrieren sich tendenziell auf nur einen oder zwei Repräsentations-Layer und werden deshalb mitunter als *Shallow Learning* (»flaches« Lernen) bezeichnet.

Beim Deep Learning werden die Repräsentations-Layer (fast immer) durch ein Modell erlernt, das *neuronales Netz* genannt wird (*Neural Network*, ab sofort mit *NN* bezeichnet). NNs sind aus buchstäblich übereinandergestapelten Layern aufgebaut. Der Begriff entstammt zwar der Neurobiologie, aber obwohl einige der grundlegenden Konzepte des Deep Learnings zum Teil inspiriert wurden durch unser Verständnis vom Gehirn, sind Deep-Learning-Modelle *keine* Nachbildungen des Gehirns. Es gibt keinerlei Hinweise darauf, dass das Gehirn irgendwelche Verfahren einsetzt, die den in modernen Deep-Learning-Modellen eingesetzten Lernmechanismen ähneln. In populärwissenschaftlichen Artikeln wird gelegentlich behauptet, Deep Learning funktioniere wie das Gehirn oder sei dem Gehirn nachgebildet worden, aber das stimmt nicht. Für Neulinge in diesem Fachgebiet wäre es verwirrend und kontraproduktiv, wenn sie annähmen, dass Deep Learning irgendetwas mit Neurobiologie zu tun hat. Vergessen Sie die Mythen und Rätsel, die um die Vorstellung »genau wie unser Gehirn« gesponnen wurden, und am besten auch alles, was Sie über hypothetische Zusammenhänge zwischen Deep Learning und Biologie gelesen haben. Für unsere Zwecke ist Deep Learning ein mathematisches Framework zum Erlernen der Repräsentationen anhand von Daten.

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by
mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

Wie sehen die von einem Deep-Learning-Algorithmus erlernten Repräsentationen eigentlich aus? Sehen wir uns doch einmal an, wie ein mehrere Layer umfassendes *Deep Neural Network* (tiefes neuronales Netz, kurz DNN) das Bild einer Ziffer umwandelt, um zu erkennen, um welche Ziffer es sich handelt (siehe Abbildung 1.5).

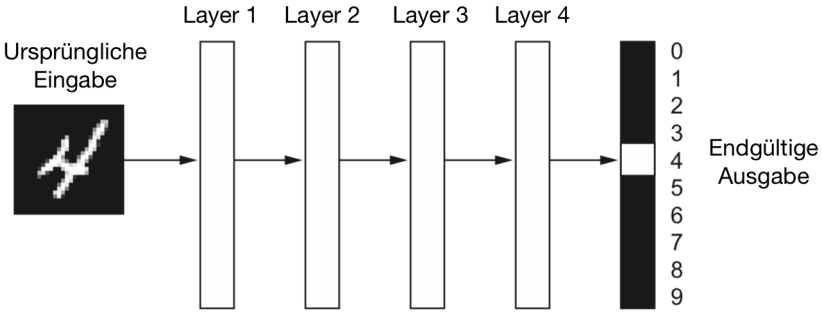

Abb. 1.5: Ein DNN zur Klassifikation von Ziffern

Der folgenden Abbildung 1.6 können Sie entnehmen, dass das Netz das Bild der Ziffer in Repräsentationen transformiert, die sich zunehmend vom ursprünglichen Bild unterscheiden und bezüglich des Endergebnisses immer aussagekräftiger werden. Stellen Sie sich ein DNN wie eine mehrstufige Operation zum Herausdestillieren von Informationen vor, bei der die Informationen aufeinanderfolgende Filter passieren und dabei immer »reiner« werden, also immer aussagekräftiger bezüglich einer bestimmten Aufgabe.

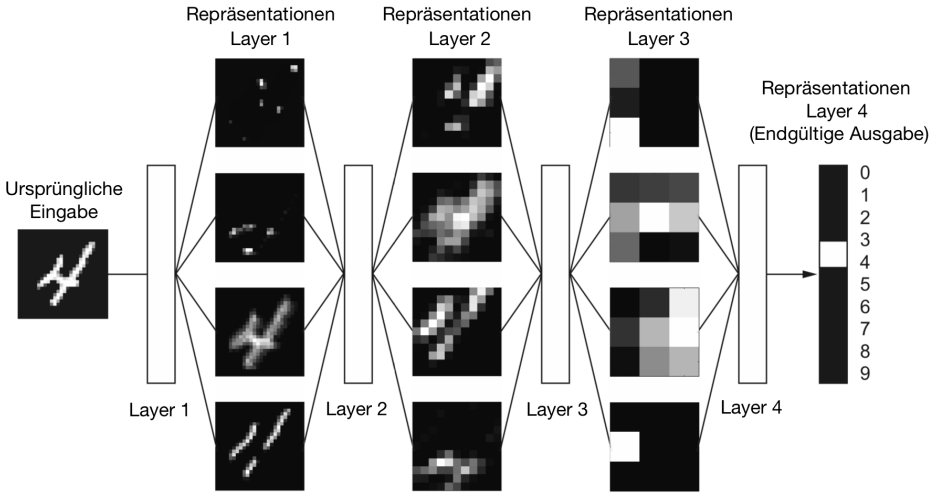

Abb. 1.6: Die von einem Klassifikationsmodell erlernten Repräsentationen

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by
mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

Technisch betrachtet ist Deep Learning ein mehrstufiges Verfahren, um Repräsentationen von Daten zu erlernen. Eigentlich eine ganz simple Idee – aber wie sich herausstellen wird, kann dieser einfache Mechanismus wahre Wunder bewirken, wenn er nur hinreichend oft durchlaufen wird.

1.1.5 Deep Learning in drei Diagrammen erklärt

Sie wissen bereits, dass es beim Machine Learning darum geht, Eingaben (wie z. B. Bilder) Zielen (wie etwa der Kennzeichnung »Katze«) zuzuordnen, indem viele Beispiele für Eingaben und Ziele betrachtet werden. Darüber hinaus wissen Sie, dass DNNs diese Zuordnung über eine Sequenz von einfachen Datentransformationen (den Layern) vornehmen und dass diese Transformationen anhand der Beispiele erlernt werden. Sehen wir uns doch einmal an, wie dieses Erlernen konkret stattfindet.

Die Angabe, was ein Layer mit den Eingabedaten anfängt, ist in den *Gewichten* des Layers gespeichert. Bei diesen handelt es sich im Wesentlichen um einen Haufen Zahlen. Man spricht hier davon, dass die von einem Layer implementierte Transformation durch die Gewichte *parametrisiert* ist (siehe Abbildung 1.7). (Die Gewichte werden manchmal auch als die *Parameter* des Layers bezeichnet.) In diesem Zusammenhang bedeutet *Learning*, einen Satz von Werten für die Gewichte aller Layer eines neuronalen Netzes zu finden, der dafür sorgt, dass das Netz alle Eingabebeispiele den zugehörigen Zielen korrekt zuordnet. Und hier liegt das Problem: DNNs können zig Millionen Parameter besitzen. Den richtigen Wert für all diese Parameter zu finden, scheint eine entmutigende Aufgabe zu sein, insbesondere in Anbetracht der Tatsache, dass das Modifizieren eines Parameters Auswirkungen auf das Verhalten aller anderen hat!

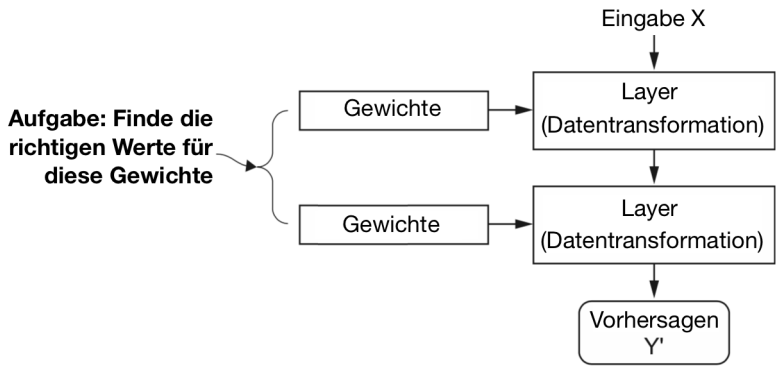

Abb. 1.7: Ein neuronales Netz wird durch seine Gewichte parametrisiert.

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by
mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

Man muss zunächst einmal in der Lage sein, etwas beobachten zu können, um es zu steuern. Zur Steuerung der Ausgabe eines neuronalen Netzes muss man messen können, wie stark die Ausgabe von dem erwarteten Wert abweicht. Diese Aufgabe erfüllt die *Verlustfunktion* des neuronalen Netzes, die manchmal auch als *Zielfunktion* bezeichnet wird. Die Verlustfunktion berechnet anhand der Vorhersage des Netzes und des tatsächlichen Zielwerts (des Werts, den das Netz ausgeben sollte) einen Verlustscore, der auf diese Weise erfasst, wie gut das Netz für dieses spezielle Beispiel funktioniert (siehe Abbildung 1.8).

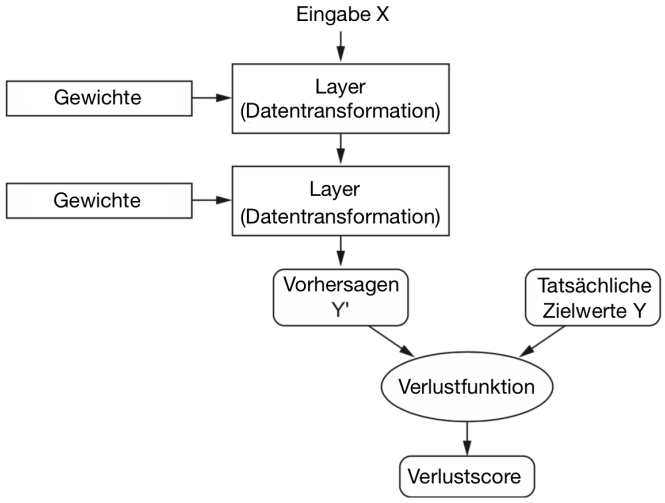

Abb. 1.8: Die Verlustfunktion bemisst die Qualität der Ausgabe eines neuronalen Netzes.

Beim Deep Learning besteht der eigentliche Trick darin, dass dieser Score als Feedback-Signal zur Feinabstimmung der Gewichte dient. Die Werte werden so geändert, dass sich der Verlustscore für das aktuelle Beispiel verringert (siehe Abbildung 1.9). Diese Anpassung ist die Aufgabe des *Optimierers*, der einen sogenannten *Backpropagation*-Algorithmus implementiert, den Hauptalgorithmus beim Deep Learning. Im nächsten Kapitel wird die Funktionsweise der Backpropagation ausführlicher erläutert.

Zunächst werden den Gewichten zufällige Werte zugewiesen – das neuronale Netz implementiert also eine Reihe zufälliger Transformationen. Die Ausgabe ist natürlich weit von den idealen Werten entfernt, und der Verlustscore ist anfangs dementsprechend groß. Doch mit jedem weiteren vom neuronalen Netz verarbeiteten Beispiel werden die Gewichte so angepasst, dass der Verlustscore abnimmt. Hierbei handelt es sich um die *Trainingsschleife*, die, nachdem sie hinreichend oft durchlaufen wurde (typischerweise etwa zehn Mal mit jeweils Tausenden von Beispielen), Gewichte liefert, die die Verlustfunktion minimieren. Bei einem neuronalen Netz mit minimaler Verlustfunktion liegen die Ausgabewerte so nah wie

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by
mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

möglich an den Zielwerten: Man spricht in diesem Fall von einem trainierten neuronalen Netz. Tatsächlich ist es dieser einfache Mechanismus, der wahre Wunder bewirken kann, wenn er nur hinreichend oft durchlaufen wird.

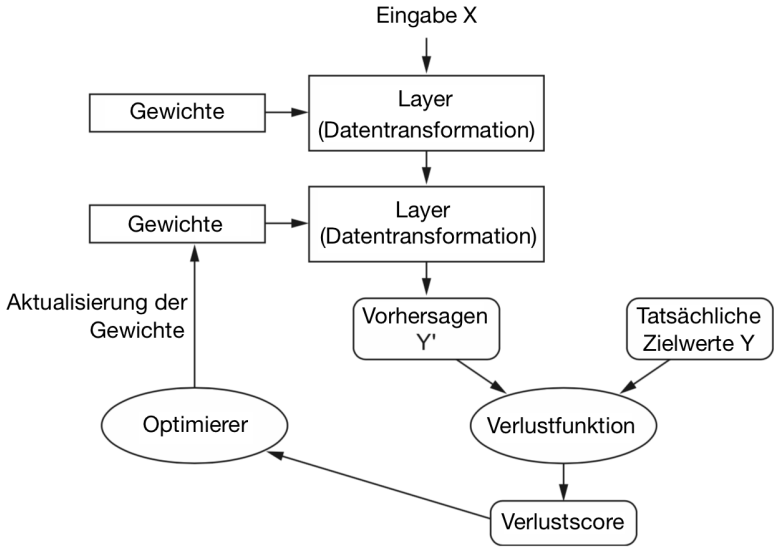

Abb. 1.9: Der Verlustscore dient als Feedback-Signal zur Anpassung der Gewichte.

1.1.6 Was Deep Learning heute schon leisten kann

Deep Learning ist zwar ein vergleichsweise altes Teilgebiet des Machine Learnings, erlangte aber erst Anfang der 2010er-Jahre größere Bedeutung. In den wenigen seither vergangenen Jahren sorgte es jedoch für eine regelrechte Revolution auf diesem Fachgebiet und erzielte erstaunliche Leistungen bei Aufgaben der Sinneswahrnehmung wie Hören und Sehen – für Menschen natürliche und intuitive Fähigkeiten, die für Maschinen jedoch lange unerreichbar waren.

Mit Deep Learning konnte auf den folgenden traditionell schwierigen Gebieten des Machine Learnings Durchbrüche erzielt werden:

- Bildklassifikation auf nahezu menschlichem Niveau
- Spracherkennung auf nahezu menschlichem Niveau
- Handschriftenerkennung auf nahezu menschlichem Niveau
- Verbesserung der Übersetzung von Fremdsprachen
- Verbesserung der Sprachsynthese
- Digitale Assistenten wie Google Now oder Amazon Alexa
- Selbstfahrende Autos auf nahezu menschlichem Niveau
- Verbesserung gezielter Werbung, wie sie Google, Baidu und Bing einsetzen

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by
mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

- Verbesserung der Suchergebnisse im Web
- Beantwortung von in natürlicher Sprache gestellten Fragen
- Ein Programm schlägt den besten menschlichen Go-Spieler

Wir sind noch immer damit beschäftigt, das Ausmaß dessen zu erkunden, was mit Deep Learning erreicht werden kann. Inzwischen wird es für eine Vielzahl von Aufgaben jenseits der Sinneswahrnehmung und des Verstehens natürlicher Sprache eingesetzt, wie etwa für das formale Schließen. Wenn sich diese Ansätze als erfolgreich erweisen, könnte das der Beginn eines neuen Zeitalters sein, in dem Deep Learning Menschen bei der Forschung, der Softwareentwicklung und vielem anderen unterstützt.

1.1.7 Schenken Sie dem kurzfristigen Hype keinen Glauben

Deep Learning hat zwar in den letzten Jahren bemerkenswerte Fortschritte erzielt, allerdings sind die Erwartungen in Bezug darauf, was auf diesem Gebiet im kommenden Jahrzehnt erreicht werden kann, viel zu hoch gesteckt. Auch wenn weltbewegende Anwendungen wie selbstfahrende Autos schon in greifbarer Nähe sind, werden viele andere wahrscheinlich noch lange unerreichbar bleiben, wie etwa glaubwürdige Sprachdialogsysteme, Übersetzungen zwischen beliebigen Sprachen oder das Verständnis natürlicher Sprache auf menschlichem Niveau. Vor allem die Berichte über allgemeine Intelligenz auf menschlichem Niveau sollten nicht allzu ernst genommen werden. Hohe Erwartungen, die kurzfristig nicht erfüllt werden, weil die Technologie noch nicht so weit ist, bringen das Risiko mit sich, dass weniger in die Forschung investiert wird und sich der Fortschritt auf diese Weise nachhaltig verlangsamt.

Es wäre nicht das erste Mal. In der Vergangenheit kam es schon zwei Mal vor, dass bezüglich der KI erst immenser Optimismus herrschte, dem Enttäuschung und Skepsis folgten, was zu einem Mangel an Fördergeldern führte. Den Anfang machte die symbolische KI in der 1960er-Jahren. In diesen frühen Tagen wurden hochgesteckte Prognosen bezüglich der KI geäußert. Marvin Minsky war einer der bekanntesten Pioniere und Verfechter der symbolischen KI. 1967 behauptete er: »Innerhalb der nächsten Generation ... wird die Aufgabe, eine ›Künstliche Intelligenz‹ zu erschaffen, im Wesentlichen gelöst sein.« Drei Jahre später, also 1970, traf er eine genauere Vorhersage: »In den nächsten drei bis acht Jahren wird es eine Maschine mit der allgemeinen Intelligenz eines durchschnittlichen Menschen geben.« Auch 2018 scheint das Erreichen dieses Ziels noch in ferner Zukunft zu liegen – so fern, dass wir nicht vorhersagen können, wie lange es noch dauern wird. Doch in den 1960er- und frühen 1970er-Jahren waren einige Experten (so wie viele Menschen heutzutage) davon überzeugt, dass der Durchbruch kurz bevorsteht. Nachdem sich die hohen Erwartungen einige Jahre später nicht erfüllt hatten, wendeten sich die Wissenschaftler von diesem Forschungsgebiet

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by
mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

ab, und die Regierung strich die Fördergelder. Das war der Anfang des ersten *KI-Winters* (eine Anspielung auf den nuklearen Winter, denn diese Ereignisse fanden kurz nach dem Höhepunkt des Kalten Kriegs statt).

Es sollte nicht der letzte KI-Winter gewesen sein. In den 1980er-Jahren entwickelte sich ein neuer Ansatz für die symbolische KI, die sogenannten Expertensysteme, die in größeren Unternehmen allmählich Fahrt aufnahmen. Einige wenige anfängliche Erfolgsgeschichten lösten eine Investitionswelle aus. Unternehmen rund um den Globus gründeten ihre eigenen KI-Abteilungen, um Expertensysteme zu entwickeln. Mitte der 1980er-Jahre gaben die Unternehmen jedes Jahr mehr als eine Milliarde Dollar für diese Technologie aus. Anfang der 1990er-Jahre stellte sich dann heraus, dass der Unterhalt dieser Systeme kostspielig war, dass sie sich nur schwer skalieren ließen und dass sie lediglich in wenigen Bereichen einsetzbar waren – das Interesse verlief im Sande. Somit begann der zweite KI-Winter.

Möglichweise sind wir gerade Zeugen eines dritten Zyklus von KI-Hype und nachfolgender Enttäuschung – und befinden uns noch in der Phase des immensen Optimismus. Am besten mäßigen wir unsere kurzfristigen Erwartungen und vergewissern uns, dass die mit den technischen Aspekten weniger vertrauten Menschen die richtige Vorstellung davon haben, was mit Deep Learning möglich ist und was nicht.

1.1.8 Das Versprechen der KI

Auch wenn die kurzfristigen Erwartungen an die KI unrealistisch sind, sieht die Zukunft langfristig doch rosig aus. Wir haben gerade erst damit angefangen, Deep Learning auf viele wichtige Aufgabenstellungen anzuwenden, die sich als umwälzend erweisen könnten – von medizinischen Diagnoseverfahren bis zum digitalen Assistenten. Die KI-Forschung hat in den vergangenen fünf Jahren, größtenteils aufgrund einer in der kurzen Geschichte der KI beispiellosen Summe von Fördergeldern, erstaunlich schnell Fortschritte erzielt. Allerdings ist nur relativ wenig von diesem Fortschritt in Form von Produkten oder Verfahren in unserem Alltag angekommen. Die meisten Forschungsergebnisse finden noch keine Anwendung oder zumindest keine Anwendung auf sämtliche Aufgabenstellungen, die sie in allen Industriezweigen lösen könnten. Ihr Arzt setzt noch keine KI ein, ebenso wenig wie Ihr Steuerberater. Und Sie selbst werden im Alltag vermutlich auch keine KI nutzen. Sie können natürlich Ihrem Smartphone einfache Fragen stellen und vernünftige Antworten erhalten. Auch die Produktempfehlungen von Amazon können ziemlich nützlich sein. Oder Sie geben bei Google Fotos den Suchbegriff »Geburtstag« ein, und augenblicklich werden Ihnen die Bilder von der Geburtstagsfeier Ihrer Tochter im letzten Monat angezeigt. Diese Technologien haben sich schon deutlich weiterentwickelt. Aber diese Tools sind noch immer nur Beiwerk des täglichen Lebens. Der Übergang dahin, dass die KI die Art und Weise bestimmt, wie wir arbeiten, denken und leben, hat noch nicht stattgefunden.

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by
mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

Heutzutage ist es schwer vorstellbar, dass die KI große Auswirkungen auf unsere Welt haben wird, weil sie bislang noch kaum im Einsatz ist. Auch 1995 wäre es schwer zu glauben gewesen, welchen Einfluss das Internet in der Zukunft haben würde. Damals konnten sich die meisten Menschen nicht vorstellen, welche Bedeutung das Internet für sie haben könnte und wie es ihr Leben verändern würde. Gleiches gilt heute für Deep Learning und KI. Aber machen wir uns nichts vor: Die KI wird sich unaufhaltsam durchsetzen. In nicht allzu ferner Zukunft wird eine KI Ihr Assistent sein, vielleicht sogar Ihr Freund. Sie wird Ihre Fragen beantworten, bei der Erziehung der Kinder zur Hand gehen und auf Ihre Gesundheit achten. Sie wird Ihnen Lebensmittel bis vor die Haustür bringen und Sie von A nach B befördern. Sie wird Ihre Schnittstelle zu einer immer komplexeren und informationsintensiveren Welt sein. Noch wichtiger ist, dass die KI der gesamten Menschheit Fortschritte ermöglichen wird, indem sie menschlichen Wissenschaftlern bei bahnbrechenden Entdeckungen auf allen Forschungsgebieten assistiert, von der Genetik bis hin zur Mathematik.

Bis es so weit ist, wird es womöglich einige Rückschläge oder vielleicht einen neuen KI-Winter geben, auf ähnliche Weise wie das Internet in den Jahren 1998 und 1999 übertrieben hochgejubelt wurde und schließlich unter einem Zusammenbruch zu leiden hatte, der zu einem Rückgang der Investitionen führte, der bis Anfang der 2000er-Jahre anhielt. Aber irgendwann wird es so weit sein. Die KI wird auf nahezu alle Vorgänge angewendet werden, die unsere Gesellschaft und unseren Alltag ausmachen, ganz so wie heutzutage das Internet.

Schenken Sie dem kurzfristigen Hype keinen Glauben, aber vertrauen Sie auf die langfristige Vision. Es wird wohl noch eine Weile dauern, bis das volle Potenzial der KI ausgeschöpft werden kann – ein Potenzial, von dem noch niemand auch nur zu träumen gewagt hat. Aber die KI wird sich unaufhaltsam durchsetzen und unsere Welt auf fantastische Weise verändern.

1.2 Bevor es Deep Learning gab: eine kurze Geschichte des Machine Learnings

Die öffentliche Aufmerksamkeit für Deep Learning und die von der Industrie getätigten Investitionen haben ein Ausmaß angenommen, das in der Geschichte der KI beispiellos ist, dennoch handelt es sich nicht um die erste erfolgreiche Form des Machine Learnings. Man kann mit Sicherheit sagen, dass die meisten heutzutage in der Industrie eingesetzten Machine-Learning-Algorithmen keine Deep-Learning-Algorithmen sind. Deep Learning ist keineswegs immer das geeignete Tool für eine gegebene Aufgabe – manchmal sind nicht genügend Daten für die Anwendung von Deep Learning vorhanden, in anderen Fällen kann ein anderer Algorithmus die Aufgabe besser lösen. Falls Sie beim Deep Learning erstmals mit Machine Learning in Berührung kommen, laufen Sie womöglich Gefahr, dass Sie

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by
mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

ausschließlich auf den »Deep-Learning-Hammer« zurückgreifen und dass plötzlich alle Machine-Learning-Aufgaben wie Nägel aussehen. Die Kenntnis anderer geeigneter Ansätze und Verfahren ist die einzige Möglichkeit, nicht in diese Falle zu tappen.

Eine ausführliche Erörterung der klassischen Ansätze des Machine Learnings geht über den Rahmen dieses Buchs hinaus, wir werden sie dennoch kurz betrachten und die Umstände beschreiben, unter denen sie entwickelt wurden. Auf diese Weise können wir Deep Learning in den Kontext des Machine Learnings einordnen und besser verstehen, wie Deep Learning entstand und warum das von Bedeutung ist.

1.2.1 Probabilistische Modellierung

Probabilistische Modellierung ist die Anwendung statistischer Prinzipien auf die Datenanalyse. Dabei handelt es sich um eine der ersten Formen des Machine Learnings, die auch heute noch sehr gebräuchlich ist. Zu den bekanntesten Algorithmen dieser Kategorie gehört der *naive Bayes-Klassifikator*.

Der naive Bayes-Klassifikator ist ein Machine-Learning-Algorithmus, der auf der Anwendung des Satzes von Bayes beruht, bei der vorausgesetzt wird, dass die Merkmale der Eingabedaten alle voneinander unabhängig sind (eine »naive« Annahme, die für die Bezeichnung namensgebend war). Diese Art der Datenanalyse ist sehr viel älter als der Computer und wurde schon Jahrzehnte vor der ersten Computerimplementierung (vermutlich in den 1950er-Jahren) von Hand angewendet. Der Satz von Bayes und die Grundlagen der Statistik entstammen dem 18. Jahrhundert, und mehr müssen Sie für die Verwendung des naiven Bayes-Klassifikators gar nicht wissen.

Die *logistische Regression* ist ein eng verwandtes Modell, das manchmal als das »Hallo Welt«-Pendant des modernen Machine Learnings bezeichnet wird. Lassen Sie sich nicht von der Bezeichnung täuschen: Die logistische Regression ist kein Regressionsalgorithmus, sondern ein Klassifikationsalgorithmus. Ebenso wie der naive Bayes-Klassifikator ist die logistische Regression sehr viel älter als der Computer, sie ist aber dessen ungeachtet dank ihrer Einfachheit und Vielseitigkeit auch heute noch nützlich. Sie ist oft das Erste, was ein Data Scientist auf eine Datenmenge anwendet, um ein Gespür für die gegebene Klassifikationsaufgabe zu erlangen.

1.2.2 Die ersten neuronalen Netze

Die ersten Formen neuronaler Netze sind vollständig von den in diesem Buch beschriebenen modernen Varianten ersetzt worden. Es ist jedoch aufschlussreich, zu wissen, welche Rolle sie bei der Entstehung des Deep Learnings spielten.

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by
mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

Die grundlegenden Konzepte neuronaler Netze wurden zwar schon in den 1950er-Jahren in Form von einfachen Modellen untersucht, es sollte jedoch noch Jahrzehnte dauern, bis dieser Ansatz richtig in Schwung kam. Es fehlte lange die Möglichkeit, große neuronale Netze effizient zu trainieren. Mitte der 1980er-Jahre änderte sich das, als mehrere Forscher voneinander unabhängig den Backpropagation-Algorithmus wiederentdeckten – eine Möglichkeit, miteinander verknüpfte parametrische Operationen mithilfe eines auf dem Gradientenabstiegsverfahren beruhenden Optimierungsansatzes zu trainieren – und ihn auf neuronale Netze anwendeten. (Wir werden diese Konzepte später im Buch noch genau definieren.)

Die erste praktische Anwendung eines *Convolutional Neural Networks* (konvolutionales neuronales Netz, kurz CNN) wurde 1989 von den Bell Labs vorgestellt. Yann LeCun kombinierte die älteren Konzepte CNN und Backpropagation und wendete sie auf die Klassifikation handgeschriebener Ziffern an. Das so entstandene Netz, das den Spitznamen *LeNet* erhielt, wurde in den 1990er-Jahren vom United States Postal Service zur Automatisierung des Lesens von Postleitzahlen auf Briefumschlägen eingesetzt.

1.2.3 Kernel-Methoden

Nachdem neuronale Netze unter den Forschern dank dieses ersten Erfolgs in den 1990er-Jahren allmählich Anerkennung fanden, erlangte ein neuer Machine-Learning-Ansatz Berühmtheit und sorgte dafür, dass neuronale Netze schnell wieder in Vergessenheit gerieten: *Kernel-Methoden*. Kernel-Methoden bilden eine Gruppe von Klassifikationsalgorithmen, von denen die *Support Vector Machine* (SVM) am bekanntesten ist. Die moderne Beschreibung einer SVM wurde Anfang der 1990er-Jahre in den Bell Labs von Vladimir Vapnik und Corinna Cortes entwickelt und im Jahr 1995 veröffentlicht.[2] Allerdings gibt es auch eine ältere, lineare Beschreibung, die bereits 1963 von Vapnik und Alexey Chervonenkis veröffentlicht wurde.[3]

SVMs versuchen Klassifikationsaufgaben zu lösen, indem sie eine geeignete *Entscheidungsgrenze* (siehe Abbildung 1.10) zwischen zwei zu verschiedenen Kategorien gehörenden Punktmengen aufspüren. Eine Entscheidungsgrenze kann man sich wie eine Linie oder Fläche vorstellen, die Ihre Trainingsdaten in zwei Gebiete unterteilt, die jeweils einer der Kategorien zugeordnet sind. Zur Klassifikation neuer Datenpunkte müssen Sie lediglich prüfen, auf welcher Seite der Entscheidungsgrenze sie sich befinden.

2 Vladimir Vapnik und Corinna Cortes, *Support-Vector Networks*, Machine Learning 20, No. 3 (1995): Seiten 273–297.

3 Vladimir Vapnik und Alexey Chervonenkis, *A Note on One Class of Perceptrons*, Automation and Remote Control 25 (1964).

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by
mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

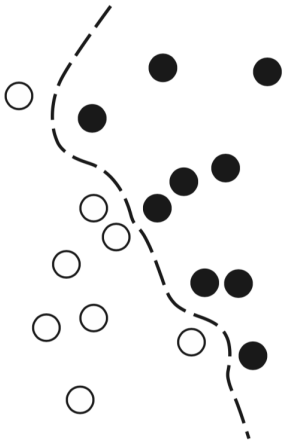

Abb. 1.10: Eine Entscheidungsgrenze

Bei der Suche nach der Entscheidungsgrenze führen SVMs zwei Schritte durch:

1. Die Daten werden auf eine hochdimensionale Repräsentation abgebildet, in der die Entscheidungsgrenze als *Hyperebene* formuliert werden kann. (Wenn die Daten, wie in Abbildung 1.10, zweidimensional sind, ist die Hyperebene eine einfache Linie.)
2. Eine geeignete Entscheidungsgrenze (eine trennende Hyperebene) wird berechnet, indem man versucht, den Abstand zwischen Hyperebene und den am nächsten gelegenen Punkten der beiden Klassen zu maximieren. Dieser Schritt wird als *Maximierung des Randbereichs* bezeichnet. Auf diese Weise lässt sich die Entscheidungsgrenze gut auf neue Punkte generalisieren, die nicht Teil der Trainingsdatenmenge sind.

Das Verfahren, Daten auf eine hochdimensionale Repräsentation abzubilden, in der die Klassifikationsaufgabe einfacher wird, mag theoretisch gut klingen, in der Praxis sind die erforderlichen Berechnungen jedoch oft kaum beherrschbar. An dieser Stelle kommt der *Kernel-Trick* ins Spiel (das entscheidende Konzept, nach dem die Kernel-Methoden benannt sind). Er funktioniert im Wesentlichen folgendermaßen: Um im neuen Repräsentationsraum eine geeignete Entscheidungsgrenze zu finden, muss man nicht unbedingt die Koordinaten sämtlicher Punkte in diesem neuen Raum berechnen. Es genügt, den Abstand von Punktepaaren in diesem Raum zu ermitteln, was sich effizient mit einer *Kernel-Funktion* erledigen lässt. Eine Kernel-Funktion ist eine Operation mit überschaubarem Rechenaufwand, die allen Punktepaaren im ursprünglichen Raum den Abstand dieser Punktepaare im neuen Repräsentationsraum zuordnet und dabei die explizite Berechnung der neuen Repräsentation vollständig umgeht. Kernel-Funktionen werden typischerweise von Hand erstellt und nicht anhand der Daten erlernt. Im Fall einer SVM wird lediglich die trennende Hyperebene erlernt.

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by
mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

Zum Zeitpunkt ihrer Entwicklung boten SVMs bei einfachen Klassifikationsaufgaben eine Leistung, die dem Stand der Technik entsprach, und gehörten zu den wenigen Methoden des Machine Learnings, die auf einer umfassenden Theorie beruhten und die einer ernsthaften mathematischen Analyse zugänglich waren. Sie waren gut verstanden und leicht interpretierbar. Aufgrund dieser angenehmen Eigenschaften wurden SVMs im Fachgebiet Machine Learning sehr beliebt und blieben es auch lange.

Aber es stellte sich heraus, dass SVMs bei großen Datenbanken schlecht skalieren und bei Aufgaben der Sinneswahrnehmung, wie z. B. der Bildklassifikation, keine besonders guten Ergebnisse lieferten. Da eine SVM dem Shallow Learning zuzuordnen ist, müssen zunächst manuell sinnvolle Repräsentationen der Daten erzeugt werden, bevor man eine SVM für Aufgaben der Sinneswahrnehmung verwenden kann. Dieser Schritt wird als *Merkmalserstellung* (engl. *Feature Engineering*) bezeichnet und ist oft schwierig und fehleranfällig.

1.2.4 Entscheidungsbäume, Random Forests und Gradient Boosting Machines

Entscheidungsbäume sind Strukturen, die Flussdiagrammen ähneln und es ermöglichen, Datenpunkte zu klassifizieren oder Ausgabewerte für gegebene Eingabewerte vorherzusagen (siehe Abbildung 1.11). Sie lassen sich leicht visualisieren und interpretieren. In den 2000er-Jahren weckten anhand von Daten erlernte Entscheidungsbäume das Interesse der Forschung. Seit 2010 werden sie oftmals anstelle von Kernel-Methoden eingesetzt.

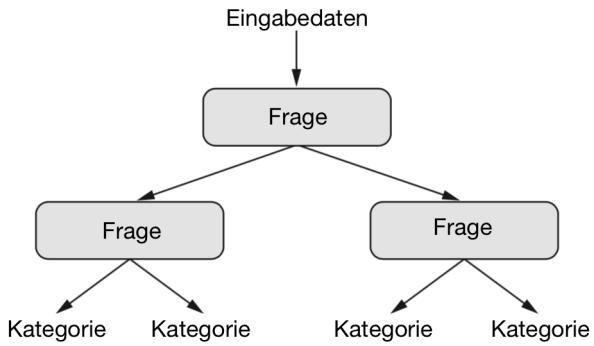

Abb. 1.11: Ein Entscheidungsbaum: Die erlernten Parameter sind die Fragen zu den Daten. Eine solche Frage könnte beispielsweise »Ist Koeffizient 2 in den Daten größer als 3,5?« lauten.

Insbesondere der *Random-Forest*-Algorithmus bot einen robusten und praxistauglichen Ansatz für das Trainieren von Entscheidungsbäumen, bei dem eine große Anzahl spezialisierter Entscheidungsbäume zum Einsatz kommt, deren Ausgaben

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by
mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

zusammengefasst werden. Random Forests sind für ein breites Aufgabenspektrum geeignet – man könnte behaupten, dass sie fast immer der zweitbeste Algorithmus für Aufgaben des Shallow Learnings sind. Als 2010 die beliebte Website für Machine-Learning-Wettbewerbe Kaggle (`https://kaggle.com`) ans Netz ging, wurden Random Forests schnell zum Favoriten dieser Plattform – bis 2014, als *Gradient Boosting Machines* diese Stelle einnahmen. Eine Gradient Boosting Machine ist, ähnlich wie ein Random Forest, ein Machine-Learning-Verfahren, das auf der Zusammenfassung vieler schwacher Vorhersagemodelle (üblicherweise Entscheidungsbäume) beruht. Es basiert auf Gradient Boosting, einer Methode zur Verbesserung beliebiger Machine-Learning-Modelle durch verschachteltes Trainieren neuer Modelle, die darauf spezialisiert sind, die Schwachpunkte der vorangegangenen Modelle zu korrigieren. Die Anwendung des Gradient-Boosting-Verfahrens auf Entscheidungsbäume führt zu Modellen, die Random Forests meist weit überlegen sind, aber ansonsten ähnliche Eigenschaften besitzen. Hierbei handelt es sich derzeit um einen der besten, wenn nicht sogar um *den* besten Algorithmus zur Handhabung von Daten jenseits der Sinneswahrnehmung. Dieses Verfahren wird, neben Deep Learning, bei Kaggle-Wettbewerben am häufigsten eingesetzt.

1.2.5 Zurück zu neuronalen Netzen

Obwohl weite Teile des Wissenschaftsbetriebs um das Jahr 2010 herum neuronale Netze nahezu vollständig mieden, arbeiteten ein paar Forscher weiter daran, und es gelangen einige wichtige Durchbrüche. Daran beteiligt waren die Forschungsgruppen um Geoffrey Hinton an der University of Toronto, Yoshua Bengio an der University of Montreal, Yann LeCun an der New York University und das IDSIA (ital. *Istituto Dalle Molle di Studi sull'Intelligenza Artificiale*, Dalle-Molle-Forschungsinstitut für Künstliche Intelligenz) in der Schweiz.

Dan Ciresan vom IDSIA konnte 2011 mit GPU-trainierten DNNs mehrere akademische Bildklassifikationswettbewerbe gewinnen – die ersten in der Praxis erzielten Erfolge des modernen Deep Learnings. Der entscheidende Schritt vorwärts gelang jedoch 2012, als Hintons Forschungsgruppe sich am jährlichen Bildklassifikationswettbewerb der ImageNet-Datenbank beteiligte. (ImageNet ist eine aus mehr als 1,4 Millionen großformatigen Bildern bestehende Bilddatenbank für Forschungszwecke.) Der ImageNet-Wettbewerb war damals bekanntermaßen äußerst schwierig, denn die zu klassifizierenden hochaufgelösten Farbfotos mussten nach dem Training mit 1,4 Millionen Bildern 1.000 verschiedenen Kategorien zugeordnet werden. Das Gewinnermodell im Jahr 2011, das auf klassischen Ansätzen des maschinellen Sehens beruhte, erzielte eine Korrektklassifikationsrate (engl. *Accuracy*, dt. auch Vertrauenswahrscheinlichkeit) von lediglich 74,3 %. 2012 erreichte ein von Alex Krizhevsky geleitetes Team, dem Geoffrey Hinton beratend zur Seite stand, eine Korrektklassifikationsrate von 83,6 % – ein bedeutender Durchbruch. Der Wettbewerb wird seither von *Deep Convolutional Neural Networks* (tiefe konvolutionale neuronale Netze, kurz DCNNs) dominiert. Im Jahr 2015 er-

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by
mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

zielte der Gewinner eine Korrektklassifikationsrate von 96,4 %, und die Klassifikation der ImageNet-Datenbank wurde als vollständig gelöste Aufgabe eingestuft.

Seit 2012 sind DCNNs für Aufgaben des maschinellen Sehens zum Algorithmus der Wahl geworden. Allgemeiner ausgedrückt, sind solche Algorithmen grundsätzlich für alle Aufgaben der Sinneswahrnehmung geeignet. Bei den größeren Konferenzen zum Thema maschinelles Sehen in den Jahren 2015 und 2016 war es fast unmöglich, einen Vortrag zu finden, bei dem es nicht in irgendeiner Form um CNNs ging. Gleichzeitig erschloss sich Deep Learning diverse weitere Aufgabenstellungen, wie z. B. die Verarbeitung natürlicher Sprache. In vielen Anwendungsbereichen hat Deep Learning SVMs und Entscheidungsbäume vollständig verdrängt. Das CERN, die europäische Organisation für Kernforschung, verwendete beispielsweise einige Jahre lang auf Entscheidungsbäumen beruhende Verfahren zur Analyse der Partikeldaten des ATLAS-Detektors am LHC (*Large Hadron Collider*). Inzwischen werden dort Keras-basierte DNNs eingesetzt, weil diese eine bessere Leistung erzielen und sich leichter mit großen Datenmengen trainieren lassen.

1.2.6 Das Besondere am Deep Learning

Der Hauptgrund dafür, dass sich Deep Learning so schnell durchgesetzt hat, ist die verbesserte Leistung, die es bei vielen Aufgabenstellungen bietet. Das ist jedoch nicht der einzige Grund. Deep Learning vereinfacht das Lösen von Aufgaben so sehr, weil es einen der entscheidenden Schritte des Machine-Learning-Workflows automatisiert: die Merkmalserstellung.

Bei früheren Machine-Learning-Verfahren, wie dem Shallow Learning, mussten die Eingabedaten in ein oder zwei aufeinanderfolgende Repräsentationsräume transformiert werden, für gewöhnlich durch einfache Transformationen wie hochdimensionale nicht lineare Projektionen (SVMs) oder Entscheidungsbäume. Aber die bei komplexeren Aufgabenstellungen erforderlichen ausgefeilteren Repräsentationen lassen sich durch diese Verfahren im Allgemeinen nicht erzielen. Ein menschlicher Benutzer musste also große Anstrengungen unternehmen, um die ursprünglichen Eingabedaten so aufzubereiten, dass sie mit diesen Verfahren verarbeitet werden konnten. Die für die Repräsentation der Daten geeigneten Layer mussten manuell konstruiert werden. Man spricht hier von der *Merkmalserstellung* (engl. *Feature Engineering*). Beim Deep Learning ist dieser Schritt hingegen komplett automatisiert. Die Merkmale werden in einem einzigen Durchgang vollständig erlernt und müssen nicht manuell erstellt werden. Diese Vorgehensweise hat den Machine-Learning-Workflow sehr vereinfacht, und in vielen Fällen wurden umständliche mehrstufige Pipelines durch ein einzelnes, einfaches und alles umfassendes Deep-Learning-Modell ersetzt.

Aber wenn das Vorhandensein mehrerer aufeinanderfolgender Layer von so entscheidender Bedeutung ist, könnte man dann nicht einfach wiederholt Shallow-

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by
mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

Learning-Methoden anwenden, um die Ergebnisse des Deep Learnings zu simulieren? In der Praxis zeigt sich, dass die Ergebnisse mehrfach angewendeter Shallow-Learning-Methoden sehr schnell deutlich schlechter ausfallen, *weil der optimale erste Repräsentations-Layer eines dreischichtigen Modells als erster Layer für ein ein- oder zweischichtiges Modell nicht optimal ist*. Beim Deep Learning ist entscheidend, dass es dem Modell ermöglicht wird, alle Repräsentations-Layer zusammen und gleichzeitig zu erlernen (man bezeichnet das als *greedy*, also gierig) anstatt der Reihe nach. Durch dieses gleichzeitige Erlernen der Merkmale werden bei der Anpassung des Modells an eins der internen Merkmale alle anderen davon abhängigen Merkmale ebenfalls automatisch angepasst, ohne dass ein menschliches Eingreifen nötig wäre. Alles wird durch ein einziges Feedback-Signal gesteuert: Alle Änderungen am Modell tragen zum eigentlichen Ziel bei. Dieses Verfahren ist sehr viel leistungsfähiger als ein massenhaftes Aufreihen von Shallow-Learning-Modellen, weil es ermöglicht, komplexe, abstrakte Repräsentationen zu erlernen, indem sie in eine lange Reihe dazwischenliegender Räume (Layer) unterteilt werden: Jeder dieser Räume unterscheidet sich von dem vorhergehenden nur durch eine einfache Transformation.

Das Lernen aus den Daten ist beim Deep Learning durch zwei wesentliche Eigenschaften gekennzeichnet: die *inkrementelle, schichtweise Vorgehensweise, bei der zunehmend komplexere Repräsentationen entwickelt werden*, und die Tatsache, dass die *dazwischenliegenden inkrementellen Repräsentationen zusammen erlernt werden*, wobei jeder Layer so aktualisiert wird, dass er den Ansprüchen der Repräsentationen der vorhergehenden und der nachfolgenden Layer genügt. Diese beiden Eigenschaften sind gemeinsam dafür verantwortlich, dass Deep Learning so viel erfolgreicher als frühere Machine-Learning-Ansätze geworden ist.

1.2.7 Der Stand des modernen Machine Learnings

Die Machine-Learning-Wettbewerbe bei Kaggle näher zu betrachten, stellt eine hervorragende Möglichkeit dar, sich ein Bild vom aktuellen Stand der Machine-Learning-Algorithmen und der verfügbaren Tools zu machen. Dank der starken Konkurrenz (bei manchen Wettbewerben gibt es mehrere Tausend Teilnehmer und Geldpreise in Millionenhöhe) und der großen Vielfalt der Machine-Learning-Aufgaben bietet Kaggle eine realitätsnahe Möglichkeit, zu beurteilen, was tatsächlich funktioniert und was nicht. Hier finden Sie die Antwort auf die Frage, welche Art Algorithmus zuverlässig Wettbewerbe gewinnt und welche Tools die besten Teilnehmer einsetzen.

In den Jahren 2016 und 2017 standen bei Kaggle zwei Ansätze im Vordergrund: Gradient Boosting Machines und Deep Learning. Gradient Boosting kommt bei Aufgaben zum Einsatz, bei denen strukturierte Daten zur Verfügung stehen. Deep Learning hingegen wird für Aufgaben der Sinneswahrnehmung genutzt, wie z. B. der Bildklassifikation. Anwender des erstgenannten Verfahrens verwenden fast

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by
mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

immer die ausgezeichnete XGBoost-Bibliothek. Die meisten Kaggle-Teilnehmer, die Deep Learning einsetzen, verwenden inzwischen die Keras-Bibliothek, denn sie ist leicht nutzbar und flexibel. Sowohl XGBoost als auch Keras bieten Unterstützung für die beiden in der Data Science beliebtesten Programmiersprachen: Python und R.

Mit diesen beiden Verfahren sollten Sie vertraut sein, wenn Sie heutzutage Machine Learning erfolgreich anwenden möchten: Gradient Boosting Machines für Shallow-Learning-Aufgaben und Deep Learning für Aufgaben der Sinneswahrnehmung. Rein technisch betrachtet bedeutet das, dass Ihnen XGBoost und Keras vertraut sein sollten – die beiden Bibliotheken, die derzeit die Kaggle-Wettbewerbe dominieren. Mit diesem Buch sind Sie Ihrem Ziel schon einen großen Schritt näher gekommen.

1.3 Warum Deep Learning? Und warum jetzt?

Die beiden grundlegenden Konzepte des Deep Learnings für maschinelles Sehen – CNNs und Backpropagation – waren 1989 bereit gut verstanden. Der LSTM-Algorithmus (*Long Short-Term Memory*, zu Deutsch etwa »langes Kurzzeitgedächtnis«), der beim Deep Learning für Zeitreihen von fundamentaler Bedeutung ist, wurde 1997 entwickelt und hat sich seither kaum verändert. Warum aber kam Deep Learning dann erst nach 2012 so richtig in Schwung? Was hat sich in den anderthalb Jahrzehnten geändert?

Es gibt drei allgemeine Faktoren, die den Fortschritt beim Machine Learning beeinflussen:

- Hardware
- Datenmengen und Benchmarks
- verbesserte Algorithmen

Dieses Fachgebiet wird nicht durch theoretische Überlegungen, sondern durch experimentelle Entdeckungen vorangetrieben, daher sind verbesserte Algorithmen nur dann machbar, wenn die entsprechenden Daten und die nötige Hardware zur Verfügung stehen, um neue Ideen auszuprobieren (oder, wie es häufig der Fall ist, ältere Ideen in größerem Maßstab umzusetzen). Anders als in der Mathematik oder der Physik ist es beim Machine Learning nicht möglich, nur mit Papier und Bleistift entscheidende Fortschritte zu erzielen. Machine Learning ist eine Ingenieurwissenschaft.

In den 1990er- und 2000er-Jahren stellten die Daten und die Hardware den eigentlichen Engpass dar. Aber in diesem Zeitraum erlebte das Internet seinen Durchbruch, und für den Bedarf des Spielemarkts wurden Hochleistungsgrafikchips entwickelt.

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by
mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

1.3.1 Hardware

Zwischen 1990 und 2010 sind handelsübliche CPUs etwa um den Faktor 5.000 schneller geworden. Dementsprechend ist es heutzutage möglich, kleine Deep-Learning-Modelle auf einem Laptop auszuführen. Vor 25 Jahren wäre das noch undenkbar gewesen.

Allerdings benötigen die beim maschinellen Sehen oder bei der Spracherkennung typischerweise eingesetzten Deep-Learning-Modelle Rechenleistung in einer Größenordnung, die ein Laptop nicht zu liefern imstande ist. In den 2000er-Jahren investierten Unternehmen wie NVIDIA und AMD Milliarden an Dollars in die Entwicklung schneller, parallel arbeitender Chips (*Graphical Processing Units*, Grafikprozessoren oder kurz GPUs), die zur Darstellung immer fotorealistischerer Videospiele dienten – preiswerte Einzweck-Supercomputer, die dafür ausgelegt sind, komplexe 3-D-Szenen in Echtzeit auf dem Bildschirm anzuzeigen. Diese Investitionen kamen schließlich auch der Wissenschaftsgemeinde zugute, als NVIDIA 2007 CUDA (`https://developer.nvidia.com/about-cuda`) vorstellte, eine Programmierschnittstelle für ihre GPU-Baureihe. Bei verschiedenen hochgradig parallelisierbaren Anwendungen, wie etwa physikalischen Simulationen, konnten einige wenige GPUs riesige CPU-Cluster ersetzen. DNNs, für die vornehmlich viele kleine Matrizenmultiplikationen erforderlich sind, lassen sich ebenfalls in hohem Maße parallelisieren. Und seit etwa 2011 programmieren einige Forscher CUDA-Implementierungen ihrer neuronalen Netze. Zu den ersten gehörten Dan Ciresan[4] und Alex Krizhevsky.[5]

Faktisch subventionierte der Spielemarkt das Supercomputing der nächsten Generation von KI-Anwendungen. Manchmal wird aus einem Spiel tatsächlich eine bedeutende Entwicklung. Eine NVIDIA TITAN X, eine Grafikkarte für Spiele, die Ende 2015 1.000 Dollar kostete, kann bis zu 6,6 TFLOPS liefern (einfache Genauigkeit): 6,6 Billionen `float32`-Operationen pro Sekunde. Das ist etwa das 350-Fache dessen, was ein moderner Laptop leistet. Mit einer TITAN X dauert es nur ein paar Tage, ein ImageNet-Modell zu trainieren, mit dem Sie vor einigen Jahren noch den ILSVRC-Wettbewerb (*ImageNet Large Scale Visual Recognition Challenge*) gewonnen hätten. Inzwischen trainieren große Unternehmen Deep-Learning-Modelle mit Clustern aus Hunderten von GPUs, die speziell auf die Anforderungen des Deep Learnings zugeschnitten sind, wie z. B. die NVIDIA Tesla K80. Die Rechenleistung, die solche Cluster bieten, wäre ohne moderne GPUs nicht möglich.

4 Siehe *Flexible, High Performance Convolutional Neural Networks for Image Classification*, Proceedings of the 22nd International Joint Conference on Artificial Intelligence (2011), `http://www.ijcai.org/Proceedings/11/Papers/210.pdf`.

5 Siehe *ImageNet Classification with Deep Convolutional Neural Networks*, Advances in Neural Information Processing Systems 25 (2012), `http://mng.bz/2286`.

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by
mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

Darüber hinaus ging die Deep-Learning-Industrie noch einen Schritt weiter und investierte in zunehmend spezialisiertere effiziente Chips für Deep Learning. Google hat auf seiner alljährlich stattfindenden Entwicklerkonferenz Google I/O 2016 das Projekt TPU (*Tensor Processing Unit*) vorgestellt, ein neues Chipdesign, das von Grund auf für die Anforderungen von DNNs ausgelegt ist und Berichten zufolge zehnmal schneller und sehr viel energiesparender ist als die leistungsfähigsten GPUs.

1.3.2 Daten

Mitunter heißt es, dass die KI eine neue industrielle Revolution einläutet. Wenn Deep Learning bei dieser Revolution die Rolle der Dampfmaschine einnimmt, dann sind die Daten die Kohle, also das Rohmaterial, das unsere intelligenten Maschinen antreibt und ohne das es nicht geht. Was die Daten betrifft, hat nicht nur das exponentielle Wachstum der Speicherkapazität in den letzten 20 Jahren (nach dem Moore'schen Gesetz), sondern auch der Aufstieg des Internets für veränderte Bedingungen gesorgt, denn nun ist es möglich, für das Machine Learning sehr große Datenmengen zu sammeln oder zu verteilen. Heutzutage verwenden große Unternehmen Datenmengen (Bilder, Videos, Sprachaufnahmen), die ohne das Internet nicht zustande gekommen wären. Die Verschlagwortung der bei Flickr gespeicherten Bilder durch die Benutzer hat sich für das maschinelle Sehen als wahre Datengoldgrube erwiesen. Gleiches gilt auch für YouTube-Videos. Und die Wikipedia ist für die Verarbeitung natürlicher Sprache eine Datenmenge von entscheidender Bedeutung.

Wenn man eine Datenmenge nennen sollte, die als Katalysator für den Erfolg des Deep Learnings gedient hat, dann die ImageNet-Datenbank, die mehr als 1,4 Millionen Bilder enthält, die von Hand einer von 1.000 Kategorien zugewiesen worden sind (eine Kategorie pro Bild). Das Besondere an der ImageNet-Datenbank ist jedoch nicht nur ihre Größe, sondern auch der dazugehörige alljährlich stattfindende Wettbewerb.[6]

Wie Kaggle seit 2010 gezeigt hat, sind öffentliche Wettbewerbe eine ausgezeichnete Möglichkeit, Wissenschaftler und Entwickler zu motivieren, ihre Forschungen voranzutreiben. Die allgemein bekannten Benchmarks, die alle Forscher gern übertreffen möchten, haben viel zu den jüngsten Erfolgen des Deep Learnings beigetragen.

1.3.3 Algorithmen

Neben der Hardware und den Daten fehlte bis Ende der 2000er-Jahre auch eine zuverlässige Möglichkeit, sehr große DNNs zu trainieren. Deshalb waren sie noch

6 Der vorhin schon kurz erwähnte ILSVRC-Wettbewerb (*ImageNet Large Scale Visual Recognition Challenge*). Siehe auch `http://www.image-net.org/challenges/LSVRC`.

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by
mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

immer nicht sehr tief und bestanden aus nur einem oder zwei Repräsentations-Layern. Dementsprechend waren sie auch nicht in der Lage, im Vergleich mit ausgefeilteren Shallow-Learning-Methoden wie SVMs und Random Forests zu glänzen. Der entscheidende Punkt war die Ausbreitung des Gradienten (engl. *Gradient Propagation*) in den tieferen Layern. Das zum Trainieren der NNs verwendete Feedback-Signal wurde mit steigender Zahl der Layer immer schwächer.

Das sollte sich 2009/2010 durch einige einfache, aber bedeutende Verbesserungen der Algorithmen ändern, die nun eine bessere Ausbreitung des Gradienten ermöglichten:

- verbesserte *Aktivierungsfunktionen* für die Layer
- verbesserte *Verfahren zur Initialisierung der Gewichte*, zunächst durch schichtweises Pretraining, was aber bald wieder aufgegeben wurde
- verbesserte *Optimierungsverfahren*, wie etwa RMSProp und Adam

Erst nachdem diese Verbesserungen es ermöglichten, Modelle mit zehn oder mehr Layern zu trainieren, zeigten sich die Vorteile des Deep Learnings.

In den Jahren 2014, 2015 und 2016 wurden weitere sogar noch fortschrittlichere Verfahren zur Verbesserung der Ausbreitung des Gradienten entdeckt, wie z. B. die Normierung der Stapel (engl. *Batch Normalization*, dt. auch als Batch-Normalisierung bezeichnet), residuale Verbindungen oder kanalweise trennbare Faltungen. Heutzutage können wir problemlos Modelle trainieren, die aus mehreren Tausend Layern bestehen.

1.3.4 Eine neue Investitionswelle

Nachdem sich Deep Learning in den Jahren 2012 und 2013 zum neuen Stand der Technik für das maschinelle Sehen und schließlich auch für alle anderen Aufgaben der Sinneswahrnehmung entwickelt hatte, nahmen die Wirtschaftsführer das ebenfalls zur Kenntnis. Daraufhin folgte eine allmählich wachsende Investitionswelle, die alles übertraf, was es in der Geschichte der KI bisher gegeben hatte.

Im Jahr 2011, unmittelbar bevor Deep Learning die Aufmerksamkeit auf sich zog, betrug das gesamte in die KI investierte Risikokapital 19 Millionen Dollar, die fast vollständig für praktische Anwendungen von Shallow-Learning-Methoden verwendet wurden. 2014 war dieser Wert auf atemberaubende 394 Millionen Dollar angewachsen. Während dieser drei Jahre wurden Dutzende Start-ups gegründet, die versuchten, vom Deep-Learning-Hype zu profitieren. Große Unternehmen wie Google, Facebook, Baidu und Microsoft hatten zwischenzeitlich Beträge in ihre internen Forschungsabteilungen investiert, die den Fluss des Risikokapitals höchstwahrscheinlich in den Schatten stellen. Es sind nur wenige konkrete Zahlen verfügbar: 2013 übernahm Google das Deep-Learning-Start-up DeepMind für angeblich 500 Millionen Dollar – die kostspieligste Übernahme eines KI-Unter-

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by
mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

nehmens der Geschichte. 2014 eröffnete Baidu im Silicon Valley ein Deep-Learning-Forschungszentrum und investierte 300 Millionen Dollar in das Projekt. Und das Start-up Nervana Systems, das Deep-Learning-Hardware entwickelt, wurde 2016 für mehr als 400 Millionen Dollar von Intel übernommen.

Machine Learning, insbesondere Deep Learning, ist für die Produktstrategie dieser Tech-Giganten inzwischen von entscheidender Bedeutung. Ende 2015 sagte Googles CEO Sundar Pichai: »Machine Learning ist ein zentrales, umwälzendes Verfahren, das wir beim Überdenken unserer Vorgehensweise zur Grundlage gemacht haben. Wir wenden es wohlüberlegt auf alle unsere Produkte an, ob Internetsuche, Anzeigen, YouTube oder Play. Und wir stehen erst am Anfang, aber Sie werden sehen, dass wir Machine Learning systematisch in all diesen Bereichen einsetzen werden.«[7]

Diese Investitionswelle hatte zur Folge, dass die Zahl der Beschäftigten, die an Deep Learning arbeiten, in nur fünf Jahren von einigen Hundert auf mehrere Zehntausende stieg und der Fortschritt der Forschung ein aberwitziges Tempo erreicht hat. Gegenwärtig gibt es keinerlei Anzeichen dafür, dass sich diese Entwicklung in absehbarer Zeit verlangsamen wird.

1.3.5 Die Demokratisierung des Deep Learnings

Zu den entscheidenden Faktoren für die Zunahme der an Deep Learning arbeitenden Beschäftigten gehört die Demokratisierung der in diesem Fachgebiet verwendeten Tools. In der Frühphase waren beträchtliche Fachkenntnisse in C++ und CUDA erforderlich, über die kaum jemand verfügte, um Deep Learning betreiben zu können. Heutzutage reichen grundlegende Python-Kenntnisse aus, um Deep-Learning-Forschung zu betreiben. Dafür verantwortlich sind vor allem die Entwicklung von Theano und später TensorFlow – zwei Python-Frameworks zur Bearbeitung von Tensoren, die automatisches Differenzieren unterstützen und die Implementierung neuer Modelle enorm vereinfachen – und die Verbreitung benutzerfreundlicher Bibliotheken wie Keras, die das Deep Learning so einfach wie das Spielen mit LEGO-Steinen machen. Nach der Veröffentlichung Anfang 2015 wurde Keras für viele neue Start-ups, Doktoranden und Forscher, die auf das neue Fachgebiet drängten, schnell zur Deep-Learning-Lösung der Wahl.

1.3.6 Bleibt es so?

Was ist das Besondere an DNNs? Weshalb sind sie offenbar das »richtige« Verfahren, in das Unternehmen investieren? Wieso befassen sich scharenweise Forscher damit? Oder ist Deep Learning nur eine kurzfristige Modeerscheinung, die nicht von Dauer ist? Werden wir in 20 Jahren noch immer DNNs verwenden?

7 Sundar Pichai bei der Bekanntgabe von Alphabets Geschäftszahlen am 22. Oktober 2015.

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by
mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

Deep Learning besitzt verschiedene Eigenschaften, die es rechtfertigen, von einer KI-Revolution zu sprechen, und wird Bestand haben. Wir werden in zwei Jahrzehnten vielleicht keine neuronalen Netze mehr verwenden, aber das, was wir dann benutzen, wird unmittelbar vom modernen Deep Learning und dessen Kernkonzepten abgeleitet sein. Diese wichtigen Eigenschaften lassen sich grob in drei Kategorien einteilen:

- *Einfachheit* – Deep Learning macht die Merkmalserstellung überflüssig und ersetzt komplexe, fehleranfällige und umständliche Pipelines durch ein einfaches, alles umfassendes trainierbares Modell, das typischerweise in Form von nur fünf oder sechs Tensoroperationen realisiert wird.
- *Skalierbarkeit* – Deep Learning lässt sich sehr gut für die Ausführung auf GPUs oder TPUs parallelisieren und nutzt so das Moore'sche Gesetz zu seinem Vorteil. Weil das Trainieren von Deep-Learning-Modellen in Form einer Stapelverarbeitung erfolgt, können zudem Datenmengen beliebiger Größe verwendet werden. (Der einzige Engpass ist die gleichzeitig verfügbare Rechenleistung, die aber dank des Moore'schen Gesetzes schnell zunimmt.)
- *Vielseitigkeit und Wiederverwendbarkeit* – Im Gegensatz zu vielen älteren Machine-Learning-Ansätzen können Deep-Learning-Modelle mit zusätzlichen Daten trainiert werden, ohne dass man wieder ganz von vorn anfangen müsste. Damit sind diese Modelle bestens für kontinuierliches Online-Learning geeignet – eine für produktiv eingesetzte sehr große Modelle wichtige Eigenschaft. Darüber hinaus lassen sich bereits trainierte Deep-Learning-Modelle für andere Zwecke erneut einsetzen und sind somit wiederverwendbar. So ist es beispielsweise möglich, ein für die Bildklassifikation trainiertes Deep-Learning-Modell als Teil einer Videoverarbeitungs-Pipeline einzusetzen. Auf diese Weise ist die geleistete Arbeit auch in zunehmend komplexeren und leistungsfähigeren Modellen von Nutzen. Außerdem ist das Deep Learning dadurch auch auf vergleichsweise kleine Datenmengen anwendbar.

Deep Learning steht erst seit einigen wenigen Jahren im Mittelpunkt des Interesses, und wir haben noch gar nicht vollständig ausgelotet, was es alles zu leisten vermag. Mit jedem Monat, der verstreicht, kommen neue Anwendungsfälle und Verbesserungen hinzu, die früher gültige Beschränkungen aufheben. Nach einer wissenschaftlichen Revolution folgt der weitere Fortschritt im Allgemeinen einer Sigmoidfunktion: Am Anfang steht eine Phase schnellen Fortschritts, die sich allmählich stabilisiert, wenn die Forscher an unüberwindbare Grenzen stoßen und die weiteren Verbesserungen nur noch in kleinen Schritten erfolgen. Das Deep Learning befindet sich Ende 2017 offenbar in der ersten Hälfte des Sigmoids, sodass in den kommenden Jahren mit weiteren bedeutenden Fortschritten zu rechnen ist.

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by
mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

Stichwortverzeichnis

0-D-Tensor 57
1-D-Tensor 57
2-D-Tensor 58
2-fache Kreuzvalidierung 134
3-D-Tensor 58
%*%-Operator 67
%<-%-Operator 97
%>% (Pipe-Operator) 82
<<- 263

A

Ableitung 75
Abstraktion 417
Adversarial Example 407
Agent 131
Aktivierung 205
Aktivierungsfunktion 47, 102
Algorithmische Subroutine 417
AMD 45
Analytical Engine 24
Anfangszustand 246
Ansätze der KI 394
Anzahl der Achsen 58
API
    funktionale 297
array_reshape()-Funktion 69
Artificial General Intelligence (AGI) 418
Atari-Spiele 131
Auswahl der Aktivierung 155
Auswahl der Verlustfunktion 155
Autoencoder 131, 373
AutoML-System 415
AWS 429

B

Babbage, Charles 24
Backpropagation 32, 38, 81
Bag-of-words 227
Baidu 48
Bayes-Klassifikator 37
Bengio, Yoshua 41, 237
Bestärkendes Lernen 131
Bewertungsmethode 153
BFGS-Algorithmus 366
bidirectional()-Funktion 278
Bidirektionales RNN 275
Bilddaten 62
Bilder fälschen 381
Bildsegmentierung 130
Binäre Kreuzentropie 103
Binärklassifikation 96, 399
binary_crossentropy 103
Boston-Housing-Price-Datensammlung 118

C

Caffe 350
Callback 315
    benutzerdefinierter 318
    TensorBoard 321
callback_early_stopping 315
callback_model_checkpoint 315
callback_reduce_lr_on_plateau 317
callback_tensorboard 321
CAM-Visualisierung 218
Carry-Spur 311
categorial_crossentropy 113
CERN 42
Chung, Junyoung 269
Ciresan, Dan 41, 45
Cloud 95
Clustering 130
CNN 401
CNTK 91
Convolutional Layer 87
Cortes, Corinna 38
CPU-Version
    installieren 427
CUDA 45
    konfigurieren 433
CUDA installieren 424
CUDA-Umgebung 425
cuDNN installieren 425

D

Daten
    Eigenschaften 139
    Redundanz 138
    repräsentative 137

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by
mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

Vektorisierung 138
Vorverarbeitung 138
Datenaugmentation 183
Decodierer 371
Deep Learning
Besonderheiten 394
Definition 29
Grenzen 406
und menschliches Gehirn 29
Vermenschlichung 406
zugrunde liegende Technologien 397
Zukunft 411
DeepDream 350
Implementierung in Keras 351
DeepMind 47
Differenziation
symbolische 81
Differenzierbarkeit 74, 396
dim()-Funktion 58
Dimensionalität 57
Dimensionsreduktion 130
Diskriminator 382, 386
dot-Operation 66
Downsampling 170
Dropout-Quote 149
Dropout-Regularisierung 148
Durchsickern von Informationen 134

E

Eck, Douglas 339
Eigen-Bibliothek 91
Einrichtung der GPU-Unterstützung 424
Embedding Layer 232, 234
Engpässe der Repräsentation 310
Ensemblemodell 331
Entropie 342
Entscheidungsbaum 40
Entscheidungsgrenze 38
Expertensystem 24, 35
Extreme Generalisierung 409

F

Faltung 166
eindimensionale 282
kanalweise trennbare 327, 402
punktweise 307
Faltungsbasis 189
Faltungskernel 167
Faltungsnetz 401
Faltungsoperation 165
Faltungsschicht *siehe* Convolutional Layer
Feature-Map 166
Feedforward-Netz 245
Fehlende Werte 140
Fehlerrückführung *siehe* Backpropagation
Feinabstimmung 200
Feynman, Richard 395
Filter 166
fit_generator()-Funktion 181
fit()-Methode 94
Fixpunkterstellung 315
Forum zum Buch 18
Fourier-Transformation 218
freeze_weights()-Funktion 197
Fully-connected Netz 399
Funktionale API 93, 297

G

Gal, Yarin 271
Gatys, Leon 359
Generalisierungsfähigkeit 142
Generative-Adversarial-Netz (GAN) 370, 381
Generator 181, 371, 382, 385
Generatorfunktion 262
Geometrische Subroutine 417
Gerichteter azyklischer Graph 306
Gewicht 31, 73, 394
Glatte Funktion 75
Globale Bibliothek 418
Glossar
Klassifikation 132
Regression 132
GloVe (Global Vectors for Word Representation) 237
Goodfellow, Ian 381
Google 47
GPU 96, 424
Gradient 76
Gradient Boosting Machine 41
Gradientenabstiegsverfahren
stochastisches 77
Gram'sche Matrix 361
Graph 394
gerichteter azyklischer 306
Graves, Alex 340
Greedy Sampling 341
GRU (Gated Recurrent Unit) 269

H

Hashkollision 230
He, Kaiming 296
Hintereinanderschaltung rekurrenter Layer 273
Hinton, Geoffrey 41, 148
history-Objekt 105
Hochreiter, Sepp 253

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by
mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

Holdout-Methode 134
HSV-Format 27
Huang, Jensen 397
Hyperebene 39
Hyperparameter 134, 156, 329
Hyperparameteroptimierung 330
Hypothesenraum 28, 88

I
image_data_generator()-Funktion 180
ImageNet 41
Inception-Modul 296, 306
Informationsleck 112
install_keras()-Funktion 427
Installation auf einem Server 426
Installation der erforderlichen Erweiterungen 423
Internet Movie Database (IMDb) 96
Ioffe, Sergey 326

K
k_gradients()-Funktion 213
Kanalweise trennbare Faltung 327, 402
Kapazität 143
Kategoriale Codierung 111
Kategoriale Kreuzentropie 113
Keras
  Einführung 89
  herunterladen 17
  installieren 92, 436
Keras und TensorFlow installieren 427
keras_model_sequential()-Funktion 93
keras_model()-Funktion 207, 298
Keras-Backend 213
Keras-Bibliothek 44, 48
Keras-Workflow 92
Kernel-Funktion 39
Kernel-Methode 38
Kernel-Trick 39
Kettenregel 81
K-fache Kreuzvalidierung 121, 136
KI
  Ansätze 394
KI (Künstliche Intelligenz)
  Definition 24
  symbolische 24
KI-Sommer 394
KI-Winter 35
Klambauer, Günter 327
Klassen
  Definition 52
Klassenbezeichnung
  Definition 52
Klassifikatorensemble 332
Kompilierung 54
Konditionierungsdaten 340
Konnektionismus 396
Konzeptvektor 372
Kreuzentropie 103
Kreuzvalidierung
  2-fache 134
  K-fache 121, 136
Krizhevsky, Alex 41, 45
Krümmung 76

L
L1-Regularisierung 147
L2-Regularisierung 147
Label
  Definition 52
Layer 394
  Definition 54
  letzter 155
  rekurrenter 87
  Wiederverwendbarkeit 311
layer_batch_normalization 326
layer_conv_1d()-Funktion 283
layer_conv_2d_transpose 383
layer_embedding 234
layer_separable_conv_2d 327
layer_simple_rnn-Layer 248
layer-Modul 54
LeCun, Yann 38, 41
Lernen
  bestärkendes 131
  selbstüberwachtes 130
  überwachtes 129
  unüberwachtes 130
Letzter Layer 155
Logistische Regression 37
Lokale Generalisierung 410
Lokales Minimum 80
Lovelace, Ada 24
LSTM (Long Short-Term Memory) 253

M
Machine Learning
  automatisiertes 415
  Definition 26
  Training 26
  Workflow 398
Maschinelles Sehen 282
Matrix 58
Maximierung des Randbereichs 39
Max-Pooling-Operation 171
Mehrfachklassifikation 109

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by
mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

Merkmalserstellung 40, 42, 140
Merkmalsextraktion 189
  mit Datenaugmentation 196
  ohne Datenaugmentation 193
Metalernsystem 417
Mikolov, Tomas 237
Minimum
  lokales 80
Minsky, Marvin 34
Mittelwert 120, 139
MNIST-Datensammlung 52
Modell 394
  als Layer 312
  als Programm 412
  Bewertung 133
  Einfachheit 147
  Generalisierungsfähigkeit 133, 142
  Hyperparameter 156
  Kapazität 143
  mit mehreren Ausgaben 302
  mit mehreren Eingaben 299
  Optimierung 142
  Regularisierung 156
  Sequential 293
  Überanpassung 155
  Vielfältigkeit 333
  Wissen 394
  Zustand 315
Modellierung
  probabilistische 37
Modul
  layer 54
Multi-Label-Mehrfachklassifikation 109, 400

N

Nervana Systems 48
Netze
  rekurrente neuronale 403
Neural-Style-Algorithmus 359
Neuronales Netz 29
  Aufbau 86
N-Gramme 226
Nichtlinearität 102
Nicht-stationäre Aufgaben 152
Nietzsche, Friedrich Wilhelm 343
Normierung 120, 139, 325
NVIDIA 45
NVIDIA CUDA Deep Neural Network Library (cuDNN) 91

O

Objekterkennung 130
Ockhams Rasiermesser 147
One-hot-Codierung 99, 228
One-hot-Hashing-Trick 230
Optimierer 32, 54, 79, 88
Optimierung 142

P

Padding 169
Parameter
  trainierbare 73
Parametrisierung 31
Pichai, Sundar 48
Pipe-Operator 53, 82
plot()-Funktion 106
Pooling 283
Portweiterleitung 434
predict()-Methode 108
Probabilistische Modellierung 37
Problem des verschwindenden Gradienten 252, 311
Profildatei 426
Programmsynthese 413
Punktweise Faltung 307

Q

Quellcode 18

R

R und RStudio
  installieren 432
Random Forest 40
Rang 58
Rang eines Tensors 58
Rechner
  einrichten 94
Redundanz 138
Regression 118, 401
  logistische 37
Regularisierung 143, 147, 156
Regularisierungsverlust 374
Rekonstruktionsverlust 374
Rekurrente neuronale Netze 403
Rekurrenter Layer 87
Rekurrentes Dropout-Verfahren 271
Rekurrentes neuronales Netz 225
Rekurrentes neuronales Netz (RNN) 245
relu-Funktion 101
relu-Operation 64
Renormierung von Stapeln 327
Repräsentation 27
Repräsentationsraum 100
Repräsentative Daten 137
Residuale Verbindungen 309

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

Response-Map 166
Reuters-Datensammlung 110
RGB-Format 27
rmsprop-Optimierer 109
RNN
  Zustand 403
ROC-AUC (Receiver Operator Characteristic Area Under Curve) 153
RStudio-Server
  zugreifen 434
RStudio-Servers auf einer AWS-GPU-Instanz 429

S

Samples
  Definition 52
Sampling 340
  stochastisches 341
Sampling-Strategie 341
scale()-Funktion 120, 139
Schachbrettartefakt 385
Schmidhuber, Jürgen 253
Schrittweite 170
Selbstüberwachtes Lernen 130
Sequential-Modell 293
Sequenzerzeugung 130
Sequenzielle Daten 62
Server-Installation 426
Shallow Learning 29
Shape 58
Sigmoidfunktion 100
Simonyan, Karen 188
Single-Label-Mehrfachklassifikation 109, 400
Sinneswahrnehmung 394
Skalar 57
Slicing 60
Softmax-Temperatur 342
sparse_categorial_crossentropy 116
Sprachmodell 340
Standardabweichung 120, 139
Steigung 75
Stetigkeit 74
Stilübertragung 359, 362
Stochastisches Gradientenabstiegsverfahren 77
Subsampling 283
Superassignment-Operator 263
Support Vector Machine (SVM) 38
SVM *siehe* Support Vector Machine
sweep()-Funktion 66
Symbolische Differenziation 81
Syntaxbaum 130
Szegedy, Christian 296, 326

T

Tangens hyperbolicus 108
Temperaturvorhersage 258
Tensor
  0-D- 57
  1-D- 57
  2-D- 58
  3-D- 58
  Attribute 58
  Definition 56
  Slicing 60
  umformen 69
TensorBoard
  Features 320
  Überwachung der Leistungskennzahlen 323
TensorFlow 91
Tensoroperation 64
  geometrische Interpretation 71
Tensorprodukt 64, 66
Testdatenmenge 53
Testmenge 133
Teststärke 154
Textdaten 226
tfruns-Paket 331
Theano 91
Tiefe eines Modells 29
Token 226
Tokenisierung 238
TPU (Tensor Processing Unit) 46
Trägheit 79
Trainierbare Parameter 73
Training 73
Trainingsdatenmenge 53
Trainingsmenge 133
Trainingsschleife 32, 73
Translationsinvarianz 165, 401
Transponierung 70
Tricks (Trainieren eines GANs) 384
Turing, Alan 25

U

Überanpassung 56, 107, 142
Überanpassung an Validierungsdaten 134
Überblick über den Installationsvorgang 423
Überwachtes Lernen 129
Umformen 70
Umgebungsvariablen 425
Unteranpassung 143
Unüberwachtes Lernen 130

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by
mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893

V
validation_data 181
validation_steps 181
Validierungsmenge 133
Vapnik, Vladimir 38
Varianz 121
Variational Autoencoder (VAE) 370, 372
Variationsverlust 365
Vektor 57
Vektordaten 61
Vektorisierung 65, 99, 138, 226
Vektorregression 132
Verbindung
  residuale 309
Verdeckte Einheit 100
Verlustfunktion 32, 54, 88
  für den Inhalt 360
  für den Stil 361
  Gewichtung 304
Vermenschlichung 406
Verschachtelte K-fache Kreuzvalidierung 137
Videodaten 63
Vokabular 228

W
Word2vec-Algorithmus 237
Worteinbettung 231, 237
  vortrainierte 232
Worteinbettungsraum 233
Wortvektor 231, 233

X
Xception 308
Xenakis, Iannes 338
XGBoost-Bibliothek 44

Z
Zeitpfeil 138
Zeitreihe 62
Zielfunktion 32, 88
Zisserman, Andrew 188
Zufälliges Raten 154
Zufälligkeit 342
Zustand 245
Zustand eines RNNs 403
Zustandsweitergabe 73

© des Titels »Deep Learning mit R und Keras« (ISBN 9783958458932) 2018 by
mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/893